

EDITORIAL

A demanda por serviços multimídia e o desejo de acessá-los de qualquer lugar e a qualquer momento fizeram com que as redes de faixa larga e as redes sem fio se tornassem os dois principais impulsionadores das indústrias de telecomunicações nos últimos tempos. Essa realidade, aliada à fusão que vem ocorrendo entre as áreas de telecomunicações e de informática, nos levou a abrir um espaço neste número da revista *Telecomunicações* para a publicação de versões estendidas de cinco artigos selecionados do V *Workshop* de Comunicação sem Fio e Computação Móvel (WCSF), que foi organizado pelo Instituto Nacional de Telecomunicações (Inatel) e ocorreu na cidade de São Lourenço entre os dias 27 e 30 de outubro de 2003. O WCSF teve início em 1999 e, ao longo destes anos, vem se estabelecendo como um dos principais fóruns nacionais para a divulgação de trabalhos de pesquisa nas áreas de comunicações sem fio e computação móvel. Os artigos convidados foram selecionados pelo comitê de programa do WCSF em conjunto com o corpo editorial da Revista *Telecomunicações*. Os temas tratados nestes artigos abrangem os seguintes aspectos e tipos de redes: controle de admissão de chamada para a nova geração de redes sem fio, protocolos *multicast* para redes estruturadas, redes de sensores sem fio, consumo de energia em redes *Ad Hoc* e a plataforma J2ME/CLDC. Acreditamos que esta diversidade de temas e de tecnologias de redes proporcionem ao leitor um panorama da evolução que tem ocorrido nas áreas de comunicações sem fio e computação móvel.

Em função da publicação dos artigos selecionados do WCSF, este número da Revista *Telecomunicações* sofreu uma alteração em relação à sua estrutura tradicional e não contém as seções de artigos tutoriais e de engenharia aplicada, que voltam no próximo número. Os três artigos científicos publicados neste número abordam os sistemas de transmissão OFDM (*Orthogonal Frequency Division Multiplexing*), que têm sido utilizados nos mais diversos tipos de redes de telecomunicações, o acoplamento entre as fibras ópticas e os guias de ondas ópticos, que encontram aplicação no desenvolvimento de dispositivos fotônicos integrados, e os sistemas microbolométricos, cuja aplicação no campo das telecomunicações está relacionada ao sensoriamento passivo remoto ou termosensoriamento.

Destacamos também a realização do IWT04 (*International Workshop on Telecommunications*) em Santa Rita do Sapucaí, no período de 23 a 27 de agosto próximo. Este importante evento científico está sendo organizado pelo Inatel em conjunto com a Faculdade de Engenharia Elétrica e de Computação (FEEC) da Unicamp e da SeçãoMG do IEEE. Estão previstas as apresentações de 54 artigos (selecionados dentre 99 submetidos), provenientes das mais importantes instituições de pesquisa do Brasil e do mundo, de 04 palestras plenárias e de 03 minicursos.

Para finalizar, gostaríamos de agradecer aos autores dos artigos publicados e todos os colaboradores da revista *Telecomunicações*, sem os quais esta história de sucesso não teria sido possível.

Prof. Dr. José Marcos Câmara Brito
Editor

A Call Admission Control Scheme for Next Generation Wireless Networks

Kelvin L. Dias, Stênio F. L. Fernandes and Djamel F. H. Sadok

Centro de Informática, Universidade Federal de Pernambuco
 CP 7851, Cidade Universitária, Recife-PE 50732-970, Brasil
 Phone: +55 81 3271.8430 Fax: +55 81 3271.8438
 {kld,sflf,jamel}@cin.ufpe.br

Abstract - This paper proposes a novel call admission control (CAC) scheme for wireless mobile networks. Our proposal avoids per-user reservation signaling overhead and takes into account the expected bandwidth to be used by calls handed off from neighboring cells based only on local information stored into the current cell where user is seeking admission. To this end, we propose the use of two time series-based models for predicting handoff load: the Trigg and Leach (TL), which is an adaptive exponential smoothing technique, and ARIMA (Autoregressive Integrated Moving Average) that uses the Box & Jenkins methodology. These methods are executed locally by each base-station or access router and forecast how much bandwidth should be reserved on a periodic time window basis. The two prediction methods are compared through simulations in terms of new call blocking probability and handoff dropping probability. Despite the TL method simplicity, it can achieve similar levels of call blocking probability and handoff dropping probability than those of the computational demanding ARIMA models. In addition, depending on the schemes settings, the prediction methods can grant an upper bound on handoff dropping probability even under very high load scenarios. The proposal is also improved with an adaptive approach to achieve a better bandwidth utilization.

Keywords - Call Admission Control, Wireless and Mobile Networks, Quality of Service, Scalability, Time Series Analysis

I. INTRODUCTION

The combination of Internet and mobile communications suggests that a coming trend will be an increasing demand for IP based wireless/mobile access to traditional and multimedia applications with varying quality of service (QoS) requirements. Figure 1 illustrates an envisioned scenario with heterogeneous wireless technologies integrated through IP mobility aware protocols (Mobile IPv4/IPv6, Cellular IP, Hawaii, etc.) that will seamlessly interwork with the global Internet [13], [14].

The research effort is especially challenging when dealing with provisioning of quality of service (QoS) guarantees. Users applications may experience performance degradation due to the properties of wireless channels and due to user mobility from handoffs. Handoff

in wireless/mobile networks is the mechanism that transfers an ongoing call from the current cell as the mobile station (MS) moves through the coverage area of the system. If the target cell does not have sufficient available bandwidth, the call will be dropped. From the user's point of view handoff dropping is less desirable than the blocking of a new call.

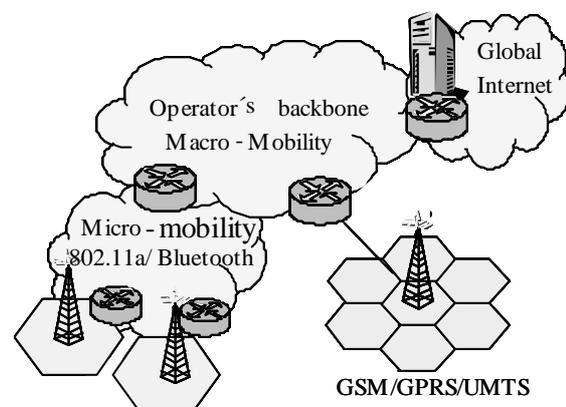


Figure 1. A scenario for all-IP mobile and wireless networks.

An important component for mobile/wireless networks is the Call Admission Control (CAC) mechanism. It must be used to address the mobility effects, accepting or rejecting new users in the network. CAC schemes not only have to ensure that the network meets the QoS of newly arriving calls if accepted, but should also guarantee that QoS of existing calls does not deteriorate.

On the other hand, Internet frameworks for QoS provisioning rely, basically, on two architectures: Integrated Services (IntServ) [15] and Differentiated Services (DiffServ) [16]. While the IntServ architecture provides strict QoS guarantees through per-user explicit signaling for CAC and reservation using RSVP (Resource Reservation Protocol), it fails in providing the scalability objectives due its reservation-based approach. The DiffServ proposal aims at providing less strict QoS guarantees through packet classification at network ingress and differentiation of the treatment according to a set of classes named PHB (Per Hop Behavior), hence offering better network scalability. The Bandwidth Broker (BB) is a network entity proposed for implementing resource management policies in the DiffServ architecture, including the CAC mechanism [17].

In wireless and mobile networks, reservation of resources is more challenging than in wired networks due

the scarcity of bandwidth in wireless links. In our opinion, a scalable QoS architecture for wireless/mobile networks should provide CAC schemes that avoid excessive per-user signaling for wireless link reservation purposes.

This paper proposes a novel call admission control (CAC) scheme for wireless and mobile networks. Our proposal avoids per-user reservation signaling overhead and takes into account the expected bandwidth to be used by calls handed off from neighboring cells based only on local information stored in the current cell where user is seeking admission. To this end, we propose the use of two time series-based models for predicting handoff load: the Trigg and Leach (TL), that is an adaptive exponential smoothing technique [9], and the Autoregressive Integrated Moving Average (ARIMA) in conjunction with the Box & Jenkins methodology [10][11]. These models indicate how much bandwidth should be reserved on a periodic time window basis. The two proposals are compared through simulations in terms of new Call Blocking Probability (CBP), Handoff Dropping Probability (HDP) and Bandwidth Utilization. Furthermore, an analysis regarding the quality of the predictions depicts that the time window prediction interval should be set carefully to avoid overestimation and so the waste of the scarce wireless bandwidth. Despite the TL method simplicity, it can achieve similar levels of call blocking probability and handoff dropping probability than those of the computational demanding ARIMA models. In addition, depending on the scheme settings, the prediction methods can grant an upper bound on handoff dropping probability even under very high load scenarios. The proposal is also improved with an adaptive approach to achieve a better bandwidth utilization.

The remainder of this paper is organized as follows. In section II, we describe the related research work. Section III gives an overview of the Trigg and Leach and ARIMA techniques for forecasting. We then present the novel CAC scheme in section IV. Performance results are presented in section V. Finally, concluding remarks are given in section VI.

II. RELATED WORK

Proposals for CAC in wireless/mobile networks present in the literature can be divided into two categories: fixed and dynamic strategies. Fixed strategies, such as the guard channel (GC) [1] scheme, give preferential treatment to handoff calls reserving a fixed number of channels exclusively for them. The advantage of this strategy is its simplicity because there is no need for the exchange of control information between base stations. However, this scheme is not flexible to handle varying traffic loads, since there is no information about current and neighboring cell's load.

Proposed dynamic reservation strategies [2],[3],[4],[5],[6],[7] extend the basic guard channel scheme according to the estimated handoff call rate derived from the number of calls in the neighboring cells and the mobility pattern of these calls to reserve bandwidth in

advance in the next cell or in a group of cells. Resource reservation can be problematic, in general, due to the possibility of poor network utilization due to unnecessary blocking of new users and can get even worse if the reservation are made in several adjacent cells. Furthermore, these schemes imply a large amount of signaling overhead.

The scheme proposed in [2] uses the aggregate history of handoffs in each cell to predict the probability a call will be handed off to a certain neighboring cell. Based on handoff prediction, the number of channels is reserved in advance. Each base station records the number of handoff failures and adjusts the reservation by changing the estimation window size. One problem with history-based schemes is the overhead to develop, store and update traffic histories for the different cells. Furthermore, due to short-term changes (e.g., diversion of traffic due to accidents) and medium-term changes (e.g., traffic re-routing during road constructions), these estimates cannot be fully reliable.

The call admission control proposed in [3] takes into consideration the number of calls in adjacent cells, in addition to the number of calls in the admission cell. The authors developed a theoretical model to compute the requirements for handoff requests in order to maintain a target handoff dropping probability. The proposed model assumes that all bandwidth requests are identical, which is not valid if multimedia services with varying bandwidth requirements are to be supported by the network.

Next, we will describe some existing research that aims at optimizing bandwidth utilization (decreasing call blocking probability), but keeping low levels of dropping probability for handoffs.

In [4] a predictive channel reservation (PCR) scheme based on mobile positioning systems (GPS - Global Positioning System) is proposed. This scheme makes predictive channel reservation for each MS based on its current position and orientation. The reservation is triggered if the MS reaches a certain threshold distance from the next cell. A reservation may be deemed invalid (false reservation) if the MS changes its moving direction. In this case, the cancellation of the reservation must be sent to de-allocate the reserved channel. Furthermore, rather than strictly mapping each reserved bandwidth portion to the MS that made the reservation, all reserved bandwidth is used as a generic pool to serve handoff requests but not new calls. When a MS arrives from a neighboring cell after a handoff, it may use bandwidth from the reserved portion if there is any available. Otherwise, the handoff connection will compete in the free bandwidth portion with other new call attempts. The HPCR (Hybrid PCR) scheme is a PCR variant, which integrates the threshold distance with GC, reserving a very small fixed portion of the bandwidth for handoffs. It was shown in [4], that this hybrid approach improves the handoff dropping probability without jeopardizing the bandwidth utilization.

The ACR (Adaptive Channel Reservation) scheme was proposed in [5] and it is based on the PCR proposal, but it uses a threshold time instead of a threshold distance

to trigger the bandwidth reservation in the next predicted cell. The authors argue that using a threshold time permits a better control of the different degrees of mobility to trigger the reservation in the next cell, avoiding waste of bandwidth due to unused reservations. For example, considering a MS located in the overlapping area of two adjacent cells with a very slow moving speed of this MS (close to 0) and requiring a channel for its call. If the PCR scheme is used, two channels (each cell has one channel occupied) will be occupied by this call, one channel is used for communication in the current cell and the other is reserved for this call in the adjacent cell because the threshold distance was reached. Since the MS of this call is almost stationary, the reserved channel may not be used for the lifetime of this call. Consequently, PCR can lead to under-utilization of wireless channels.

The PCR as well as the ACR schemes introduce a lot of signaling messages for reservation and cancellation of false reservations. Moreover, the reservations can decrease the dropping probability at the expense of increasing the blocking probability, what may give rise to poor network utilization. The use of GPS for predicting user mobility is also advocated in proposals [6],[7]. While such dynamic reservation-based schemes have demonstrated significant performance advantages over well engineered guard channels, the per-user dynamic reservation approach place computation and communication burdens on the network's infrastructure which increases with the numbers of users and handoffs. Hence, the scalability and applicability of such solutions to future micro and pico-cellular networks is not well established.

A similar approach to ours is proposed in [8]. The authors proposed a local predictive resource reservation for handoff based on the Wiener process (a Markov process where only the present value is relevant for predicting the future) and a methodology for granting an upper bound on HDP. To grant an upper bound on HDP, the amount of resource that must be reserved for future handoff demands should be set to the upper limit of the confidence interval for the predicted handoff load. In addition, the authors also use an ARIMA prediction method and show that the Wiener prediction obtained quite similar results for predicting the handoff demand based on traces collected from a single cell simulation scenario. The limited results obtained for the CBP and HDP metrics were depicted only for the Wiener-based proposal. The lack of performance results in terms of CBP and HDP for their ARIMA-based prediction seems to be justified by the very similar results obtained from the comparative trace analysis with the Wiener-based method conducted in that paper. As it will be shown in our paper, applying the methodology suggested in [8] could lead to bandwidth overestimation for handoffs. Furthermore, our ARIMA-based proposal differs significantly from that in [8] because we did not adopt the upper limit of the predicted handoff confidence interval to reserve bandwidth. Instead, we suggest directly the use of the predicted value by choosing an appropriate prediction time window size to avoid unnecessary reservations of the scarce wireless bandwidth.

III. FORECASTING PROCEDURES

In this section we present a short description of the forecasting procedures used to evaluate the traffic load arriving at each cell.

A time series can be defined as a realization of a stochastic process. Time series may enfold features such as trends and seasonality and one of the purposes of its analysis is the generation of forecast of future values. This procedure normally requires that time series present some kind of regularity in its behavior. Usually, future values are predicted based on past values, because a steadiness is assumed. This regularity in time series can be expressed through the concept of stationary time series[10]. Therefore, forecasting techniques are based on the idea that future can be predicted by discovering specific patterns of events in the past. Using time series modeling and analysis to predict bandwidth requirement in a computer network environment has lately become a useful and widespread tool. Researchers in the networking field are increasingly adopting modeling techniques widely used by econometricians and statisticians [12].

III.1 EXPONENTIAL SMOOTHING AND VARIANTS

Exponential smoothing techniques have long been the methods of choice for univariate forecasting due to its accuracy and ease of use. They have become increasingly accepted because of their effortlessness and overall performance. It is highly recommended for short-term prediction. Among the simplest methods is the ordinary (simple) exponential smoothing, which assumes no trend and no seasonality whereas Trigg and Leach procedure could be seen as its adaptive approach.

III.1.1 SIMPLE EXPONENTIAL SMOOTHING

Let Y_t denote a univariate time series. Simple exponential smoothing assumes that the forecast \hat{Y} for period $t+h$ is given by a variable level \hat{a} at period t

$$Y_{t+h} = \hat{a}_t, \quad (1)$$

which is recursively estimated by a weighted average of the observed and the predicted value for Y_t .

$$\hat{a}_t = \alpha Y_t + (1-\alpha)\hat{Y}_t \quad (2)$$

$$\hat{a}_t = \alpha Y_t + (1-\alpha)\hat{a}_{t-1} \quad (3)$$

where $0 < \alpha < 1$ is known as the smoothing parameter (constant). The main drawback of this technique is the choice of the smoothing parameter since setting it close to 1 could give rise to a highly reactive model. On the contrary, choosing the smoothing constant close to 0 could lead to an insensitive model.

III.1.2 ADAPTIVE EXPONENTIAL SMOOTHING: TRIGG AND LEACH

In order to assist the selection of α and to improve awareness capability of the predictor, a number of

adaptive methods have been recommended in the literature. The most representative and widely used is the Trigg and Leach [9] technique. Its main advantage relies on the fact that there is no need to specify the smoothing parameter previously. Trigg and Leach procedure can regulate the smoothing constant α whenever a change occurs in the time series basic structure. Let α_{t+1} be the one-step ahead smoothing parameter. So, the prediction in $t+1$ for the level is

$$\hat{a}_{t+1} = \alpha_t Y_t + (1 - \alpha_t) \hat{Y}_t \quad (4)$$

$$\alpha_{t+1} = \left| \frac{E_t}{M_t} \right| \quad (5)$$

where

$$E_t = \beta \varepsilon_t + (1 - \beta) E_{t-1}, \quad (6)$$

$$M_t = \beta |\varepsilon_t| + (1 - \beta) M_{t-1} \quad (7)$$

and $\varepsilon_t = Y_t - \hat{Y}_t$ (prediction error at t).

Values close to zero point out a well-controlled prediction system (smaller prediction errors) whereas values near to the unity indicate an out of control prediction system (huge prediction errors). It is important to emphasize that α_{t+1} allow the system to reconcile by not being too reactive to changes. But most importantly, α_t will vary based on variations in the data pattern.

III.1.2.1 TRIGG AND LEACH UPPER CONFIDENCE BOUND

In order to offer statistical guarantees regarding the worst-case handoff dropping probability (HDP) for the next time interval, we may use predicted value as the upper confidence bounds for that predicted value as suggested in [8] for the Wiener process. For example, if the network operator has to guarantee a maximum target handoff dropping probability of 5%, the reserved bandwidth ψ will be set to the 95% upper confidence bounds of the forecasted bandwidth requirements for handoff calls $E(\Omega)$. This way, we can determine a level L such that $Prob(\Omega \leq L) = 1 - HDP$. This level L is called $(1 - HDP) * 100\%$ upper confidence bound for Ω . This value is given by:

$$\psi = E(\Omega) + Z_\gamma \sqrt{\left(\frac{\alpha}{2 - \alpha} \sigma^2 \right)}, \quad (8)$$

where Z_γ is the q -quartile of the standard Normal distribution of $N(0,1)$, α is the smoothing parameter, and σ^2 the sample variance.

III.2 ARIMA MODELS AND THE BOX & JENKINS METHODOLOGY

There are some classical approaches for modeling stationary time series. Models for stationary processes are the Autoregressive (AR), the Moving Average (MA) and

the Autoregressive Moving Average (ARMA). Taking a time series $\{X_t\}$, which is stationary and with nonseasonal patterns, if it follows an autoregressive process of order p , denoted by $X_t \sim AR(p)$, then $\{X_t\}$ is given by

$$X_t = c + \phi_1 X_{t-1} + \phi_2 X_{t-2} + \dots + \phi_p X_{t-p} + \varepsilon_t, \quad (9)$$

where $c, \phi_1, \phi_2, \dots, \phi_p$ are unknown parameters, the ϕ_i being called autoregressive parameters, and ε_t is a white noise process [10]. The term Moving Average comes from the fact that $\{X_t\}$ is built from a weighted sum, similar to an average, of the most recent values of ε , and then it can be expressed as

$$X_t = \mu + \varepsilon_t + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \dots + \theta_q \varepsilon_{t-q}. \quad (10)$$

$\mu, \theta_1, \dots, \theta_q$ are unknown parameters, the θ_i being called moving average parameters, and ε_t is a white noise. If X_t follows a moving average process of order q , it is denoted by $X_t \sim MA(q)$. It is possible to build models that pursue simultaneously autoregressive and moving average expressions. One example is a time series $\{X_t\}$ that follows an autoregressive process with moving average terms, denoted $X_t \sim ARMA(p, q)$, given by

$$X_t = c + \phi_p X_{t-p} + \varepsilon_t + \theta_q \varepsilon_{t-q}, \quad (11)$$

where c, ϕ_i and θ_i are unknown parameters, the ϕ_i being the autoregressive parameters and the θ_i being the moving average parameters. This is an autoregressive moving average process of order (p, q) .

It is possible that the traffic load presents some non-stationary patterns, which induces the use of classical approaches for modeling them, such as the Autoregressive Integrated-Moving Average (ARIMA) and the Seasonal Autoregressive Integrated-Moving Average (SARIMA). Another approach is to use some kind of conversion in order to make it stationary. For example, one can take differences, logarithms or squared roots of the observations. A traditional procedure is to use a class of transformations called the Box-Cox transformation [10].

Particularly, processes that, after the application of a finite number d of differences, reduce to ARMA models are called $ARIMA(p, d, q)$ models. The application of difference to the time series is a method to transform a non-stationary time series to a stationary one. An $ARIMA(p, d, q)$ model can be represented by

$$\Delta^d X_t = \mu + \phi_p \Delta^d X_{t-p} + \varepsilon_t + \theta_q \varepsilon_{t-q}, \quad (12)$$

where the ϕ_i are the autoregressive parameters, the θ_i are the moving average parameters and Δ^d indicates that the order of differentiation is d .

Taking a close look at the equation 3 it is necessary to find a way to estimate the values of $\theta \equiv (c, \phi_1, \phi_2, \dots, \phi_p, \theta_1, \theta_2, \dots, \theta_q)$, known as the vector of population parameters, on the basis of observations on

$\{X_T\}$. A usual inference technique on which estimation could be based is Maximum Likelihood (ML). Given the sample of size T, the first step is to calculate the likelihood function (LF), $L(\theta; x)$. This function can be found by calculating a probability density $f_{x_T, x_{T-1}, \dots, x_1}(x_T, x_{T-1}, \dots, x_1; \theta)$ that is strong related to the assumption that the particular distribution for the white noise process ε_t assumes a Gaussian white noise form, i.e., $\varepsilon_t \sim i.i.dN(0, \sigma^2)$. So, the maximum likelihood estimate of θ is the value for which this sample is most likely to have been observed, that is $\theta = \arg \max L(\theta; x)$, $\theta \in \Theta \subset \Re$. It is a common sense to use the reduced and conditional log-likelihood $l(\theta; x) \propto \ln L(\theta; x)$, where the LF has the form

$$L(\theta; x) = f_{x_1}(x_1; \theta) \prod_{t=1}^T f_{x_t | x_{t-1}}(x_t | x_{t-1}; \theta).$$

For example, it is easy to show that the conditional log-likelihood function for a Gaussian ARIMA(p,0,q) process is

$$l(\theta; x) = \frac{-T}{2} \log(2\pi) - \frac{T}{2} \log(2\sigma^2) - \sum \frac{\varepsilon_t^2}{2\sigma^2}, \quad (13)$$

where $\varepsilon_t = x_t - c - \phi_1 x_{t-1} - \Lambda - \phi_p x_{t-p} - \theta_1 \varepsilon_{t-1} - \Lambda - \theta_q \varepsilon_{t-q}$.

An alternative solution for (5) could be performed by solving the system of equations given by $\nabla l(\hat{\theta}) = 0$, usually referred to as likelihood equations. In both cases, there is no closed-form or explicit solution and therefore numerical maximization must be used. The idea would be to make a number of distinct guesses for θ , and try to infer the value of $\hat{\theta}$ for which $l(\theta; x)$ is largest. There are several algorithms for numerical maximization or optimization procedures. For instance, one could use Grid Search, Steepest Ascent, Newton-Raphson, Davidon-Fletcher-Powell or Broyden-Fletcher-Goldfarb-Shanon (BFGS) methods. In this work, the BFGS algorithm was used.

Needless to say that is indispensable a formal procedure to estimate the best model given a number of observations. This leads to a discussion of stochastic model building where Box & Jenkins methodology is widely used to discover models from the series, estimate their parameters and then evaluate the adequacy of the model's fit to the experimental data. The Box & Jenkins methodology tries to provide a flexible procedure so that one may obtain high-quality and suitable models. The methodology consists of three basic stages: Identification, Parameter Estimation and Diagnostic Checking. We refer the reader to Harvey [11] for a more complete explanation related to the Box & Jenkins procedure.

In the first stage, a tentative model is normally selected based on the sample autocorrelation function or the correlogram, which tries to identify the p and q orders for the ARIMA process. Given a time series, the first stage may recommend a number of specifications (i.e., p and q orders), each of which satisfies some diagnostics checks. For that reason, some kind of measure of goodness of fit is required to decide on the best models

presented. There are a number of model selection criteria where the decision rule is to select the model that minimizes some variable. The Akaike Information Criterion (AIC) has the following form:

$$AIC = -2 \log L(\theta; x) + 2n, \quad (14)$$

where n is the number of parameters ($n = p + q$). The ML estimation previously explained was performed during the second stage.

In this work, we automate the B&J methodology in order to identify, estimate and perform the diagnostic check to the handoff load on every cell on a cellular network. We used a sample time interval of 30 or 60s and collected the first 30 samples (called the training period) before starting the automated B&J procedure. This quantity is a sufficient amount of samples to achieve convergence to the ML estimative. After the training period, for each new handoff load measured in each cell, during a chosen sample time interval, we performed the whole B&J procedure all over again.

IV. THE PROPOSED CAC

Our novel CAC estimates the total amount of required bandwidth for future handoff calls using TL or ARIMA. The process for predicting the required bandwidth for handoff calls is local, that is, the base station uses only local information (collected bandwidth due to handoffs) that serves as the input for the prediction method without exchange of messages among neighboring cells to this end. Suppose that a base station knows the amount Ω of required resources for handoff calls at the current time t. The amount of resources required for handoffs $E(\Omega)$ at a future time $t + \Delta t$ can be predicted based on the current Ω and its predicted value from the previous time interval $t - \Delta t$.

The novel CAC should determine whether the admission cell has sufficient bandwidth to support the user requirements and takes into account the predicted handoff load for that cell. Let ψ (the reserved bandwidth) be the upper confidence bound for the expected bandwidth due to handoff calls $E(\Omega)$ for the next prediction interval. The reserved bandwidth can also be the actual forecasted value from a chosen time series model. The following condition must be met:

$$\sum_{i=1}^N Bi + B + \psi \leq C \quad (15)$$

This equation verifies whether the admission cell has sufficient bandwidth to support the new request. N is the number of existing connections, C is the wireless link capacity and Bi is the bandwidth being used by the ith connection in that cell. B is the bandwidth required by the newly requested connection. At the start of each interval, a new ψ is used to control the admission decision. Upon each handoff arrival in a cell, during a prediction interval, the current ψ is decreased by the MS's bandwidth that has arrived until it reaches a null value or a new prediction interval is initiated.

V. PERFORMANCE ANALYSIS

The simulated model consists of a cellular network with 19 hexagonal cells as depicted in Figure 2. In order to avoid the border effects, when a MS moves out the system this MS will be wrapped around to re-enter the system from the other side. Such a toroidal arrangement is an efficient way to approximately simulate very large systems [3], [5]. In this paper, the unit of bandwidth is called bandwidth unit (BU), which is assumed to be the required bandwidth to support a voice connection as in [2], [7]. Each cell is assumed to have a fixed link capacity of 100BUs. The traffic model used is similar to the one used in [2], [7]. Call requests are generated according to Poisson distribution with rate λ (call/cell/second) in each cell. The simulated traffic consists of users with bandwidth requirements of 1 BU (voice) and 4 BUs (video) with probabilities R_{vo} and $1-R_{vo}$, respectively, where R_{vo} is also called the voice ratio as in [2]. In our simulations R_{vo} is set to 0.7, that is, 70% of voice traffic and 30% of video traffic. The lifetime of each call is exponentially distributed with mean 180s [4], [5], [7].

Upon each new call request or handoff call, the user chooses a moving direction among six probable target cells. At any time, while crossing a cell the MS can change its moving direction with probability equal to 50%. If a MS changes its moving direction, a new target cell is randomly selected (uniformly distributed) as well as a new residence time is chosen. The time that a call spends in a cell prior to handoff to another cell (residence time) is exponentially distributed with mean 60s.

V.1 SIMULATION RESULTS

The metrics of interest in this paper are: (1) handoff dropping probability (HDP) defined as the ratio of the number of handoff calls dropped to the total number of handoff call attempts; (2) call blocking probability (CBP), that is, the ratio of the number of new calls blocked by the network to the number of new call requests; and (3) bandwidth utilization.

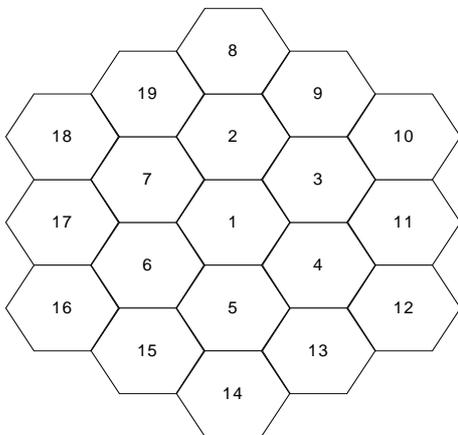


Figure 2. Simulated Cellular Topology.

Unless otherwise stated, the models are labeled “M-B-T” in graphs, where M represents the model adopted for prediction (TL or ARIMA), B is the reserved bandwidth

type which may be based either on the predicted value (Pred) or on the upper confidence bound (CI) for that predicted value. T is prediction interval (30 or 60s).

Figure 3 and Figure 4 depict the CBP and HDP comparison for the models using a prediction interval of 30s. Both models achieved similar levels of CBP. The HDP comparison shows that TL achieved a slightly greater HDP than ARIMA’s in higher loads. This scenario indicates that simplistic TL method can achieve satisfactory levels of prediction as compared to the ARIMA model.

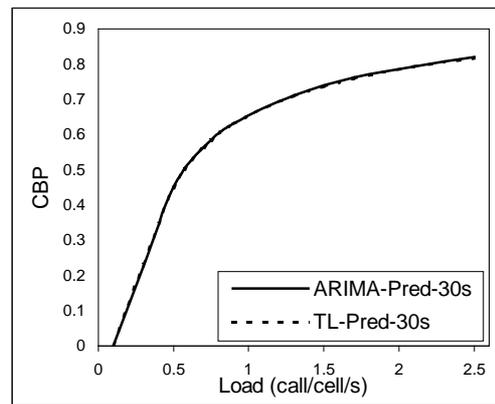


Figure 3. CBP - Prediction Interval: 30s; Reserved Bandwidth Type: Predicted Value.

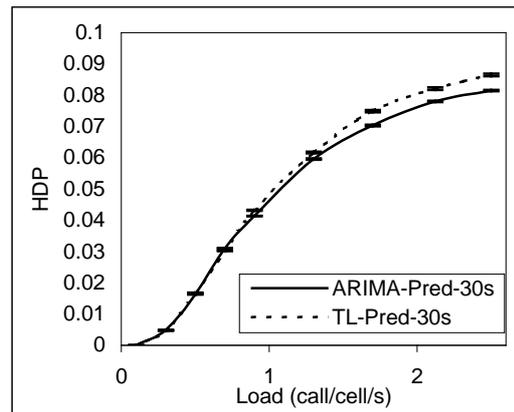


Figure 4. HDP - Prediction Interval: 30s; Reserved Bandwidth Type: Predicted Value.

In order to evaluate the proposal of using the upper confidence bound for the predicted value as the amount of bandwidth that should be reserved on each cell to guarantee a maximum target HDP during the cell overload, Figure 5 and Figure 6 show results considering the upper confidence bound for a 95% confidence level (CI). Hence, it is expected that the worst case HDP be inferior to 5%. As can be seen, the ARIMA’s HDP is better than TL’s.

However, TL’s HDP was kept below the maximum target HDP of 5%. Moreover, the smallest HDP for ARIMA is achieved at the expense of a greater CBP, which generated more blockings of new calls than TL’s (Figure 5) and, consequently, providing bandwidth underutilization as it is depicted in Figure 7, where the bandwidth utilization for TL outperforms the one for ARIMA.

In order to verify our argument that the approach of using the upper confidence bound for the predicted load may cause overestimation for bandwidth reservations, Figure 8 and Figure 9 depict comparisons between the 95% upper confidence bounds for the forecasted value and the actual handoff demand collected during the simulations for TL and ARIMA, respectively. These graphs were based on traces obtained from the same simulation as that of Figure 5 and Figure 6, considering only the load 1.7 (call/cell/second). They refer to bandwidth due to handoffs into the central cell (cell 1 depicted by Figure 2) in our topology of 19 cells. It is easy to see that using the upper confidence bound of the predicted value ARIMA models may overestimate the bandwidth needs for reservations. Hence, it is important to take into account the tradeoff between the HDP and the bandwidth utilization.

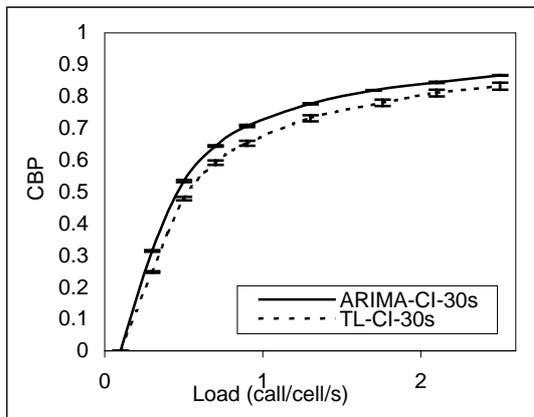


Figure 5. CBP - Prediction Interval: 30s; Reserved Bandwidth Type: Upper Confidence Bound Value

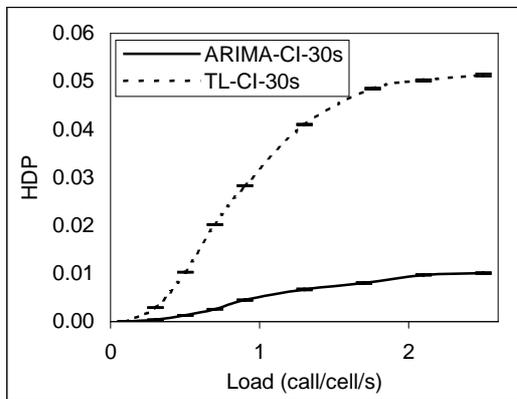


Figure 6. HDP - Prediction Interval: 30s; Reserved Bandwidth Type: Upper Confidence Bound Value.

We believe that a more interesting prediction approach is to adopt an adequate prediction interval and the predicted value forecasted by the method (TL or ARIMA). By regulating the prediction interval (i. e., the time window adopted for making forecasts), it may be possible to achieve the desirable level of HDP without jeopardizing the bandwidth utilization.

In order to check if a different prediction interval could provide a smaller and controlled HDP Figure 10 and Figure 11 show the results for TL and ARIMA using different prediction intervals and the predicted value

instead of the upper confidence bound for that value to reserve the wireless bandwidth. When the methods use 60s as the prediction interval, both methods kept HDP below 5%. The ARIMA model achieves the smallest HDP in both scenarios (i.e., with 30 and 60s). Again, choosing the appropriate prediction interval is a tradeoff between the desirable HDP and bandwidth utilization.

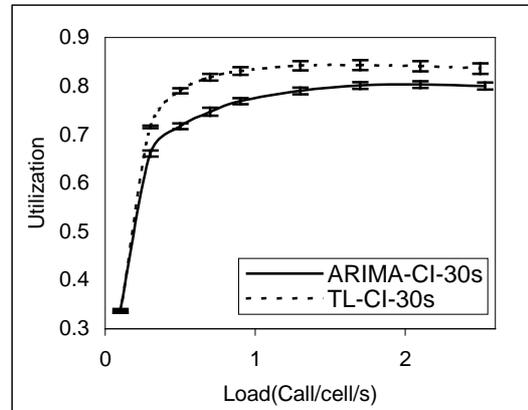


Figure 7. Utilization - Prediction Interval: 30s; Reserved Bandwidth Type: Upper Confidence Bound Value

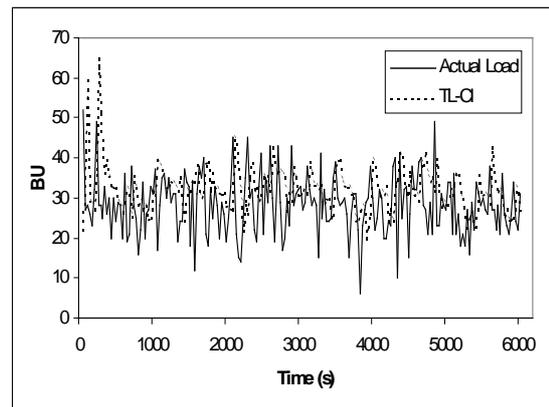


Figure 8. Actual and upper confidence bound for the predicted handoff load (TL).

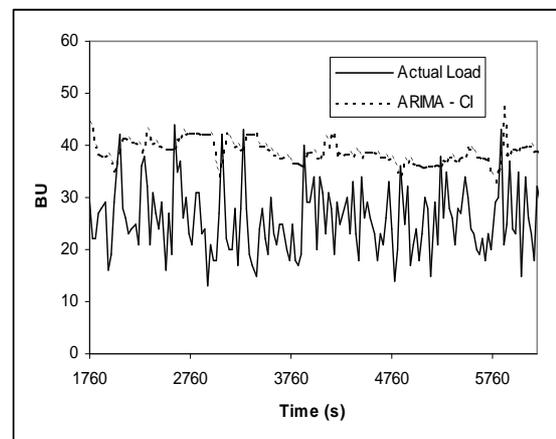


Figure 9. Actual and upper confidence bound for the predicted handoff load (ARIMA).

In order to evaluate the impact of a residence time following a Pareto Probability Distribution Function (PDF), we only used the TL model. We analysed such consequences on the effective utilization, HDP and CBP metrics using different prediction intervals and also comparing residence times following an Exponential PDF. Figure 12 and Figure 13 show the HDP and CBP behaviour when varying the prediction intervals and the residence time PDF (e.g. Pareto and Exponential). One should notice that a Pareto residence time point out to less HDP. In such situation, there are a large number of handoff events (due to small residence times) since the Pareto is a heavy tailed PDF. Hence, this behaviour leads to more blocking of new calls.

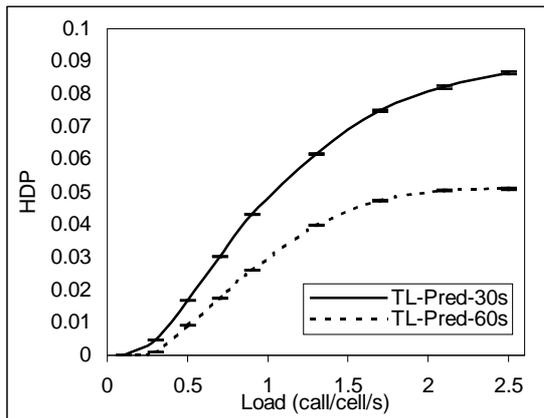


Figure 10. HDP- Prediction interval: 30s and 60s (TL).

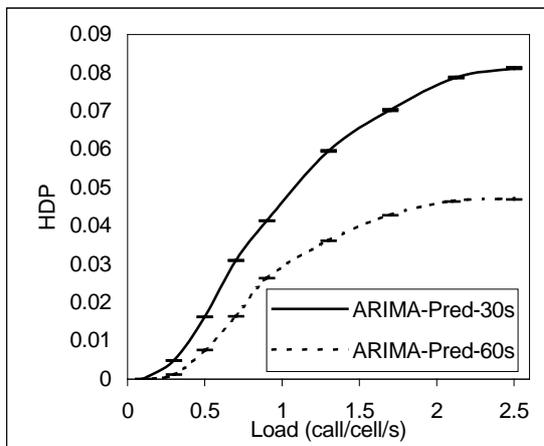


Figure 11. HDP – Prediction interval:30 and 60s (ARIMA).

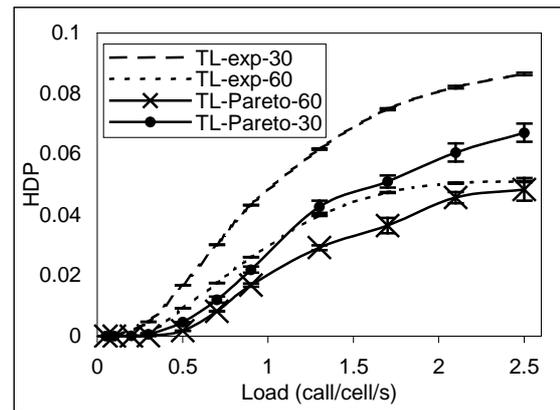


Figure 12. HDP – Residence Time: Exponential vs. Pareto

Although our methodology guarantees a maximum HDP by choosing a proper prediction interval, its optimum value depends on the traffic characteristics and also on the user’s mobility pattern. The choice of a preset value could lead to poor effective utilization since the predicted load to be reserved in each cell is directly proportional to such prediction interval. So, in order to assist to the right selection of this interval, we suggest the deployment of an adaptive scheme, which uses the measured HDP value in each cell as the main selection criteria. We describe the algorithm below:

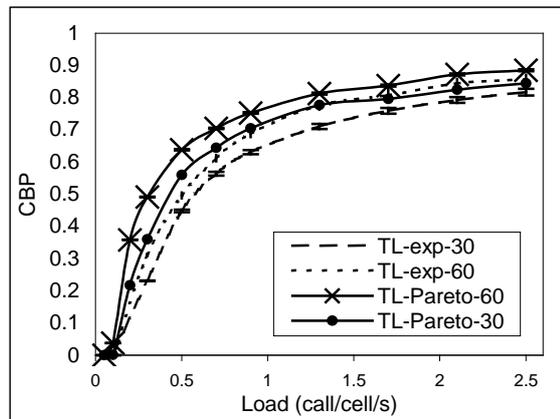


Figure 13. CBP – Residence Time - Exponential vs. Pareto

Variable description

- *Max_HDP* -> maximum value for the HDP. Beyond this threshold the algorithm must increase the prediction window.
- *Min_HDP* -> minimum value for the HDP. Below this threshold the algorithm must increase the prediction window.
- *Max_Interval* -> maximum value for the prediction interval.
- *Min_Interval* -> minimum value for the prediction interval.

Do (every prediction interval in each cell):

If ($Measured_HDP \geq Max_HDP$ &&
 $Prediction_Interval < Max_Interval$)
Increment $Prediction_Interval$;
If ($Measured_HDP < Min_HDP$ &&
 $Prediction_Interval > Min_Interval$)
Decrement $Prediction_Interval$;

Figure 14 and Figure 15 present the HDP and bandwidth utilization results from the adaptive method when using a residence time following an Exponential distribution. The scheme’s parameters are described in Table 1. The adaptive scheme is compared to the fixed one considering the prediction intervals equal to 30s e 60s. The value chosen for incrementing the interval prediction size is 5, on the other hand, the value for decrementing the interval prediction size is 1. These values were chosen in order to provide a fast recover during congestion periods as well as to avoid sudden dropping of handoffs while testing the adequate prediction interval.

Max_HDP	5%
Min_HDP	2%
Max_Interval	60s
Min_Interval	10s
Initial Prediction Interval	30s

Table 1. Parameters of the Adaptive Scheme.

Note that the adaptive scheme surpasses the fixed schemes in terms of bandwidth utilization (Figure 15) for lower loads. On the other hand, the adaptive scheme’s HDP obtained the highest values for the lower load scenarios, but kept the target HDP below 5% for higher loads as depicted in Figure 14. The HDP is also kept below the maximum permitted HDP for the fixed scheme that used 60s as the prediction interval, but its bandwidth utilization is similar to adaptive scheme’s utilization for higher loads.

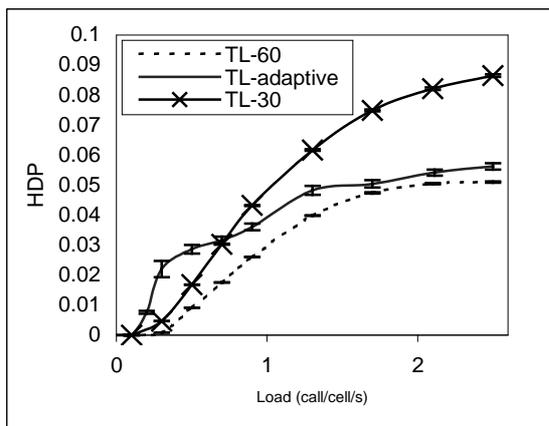


Figure 14. HDP: Adaptive vs. Fixe.

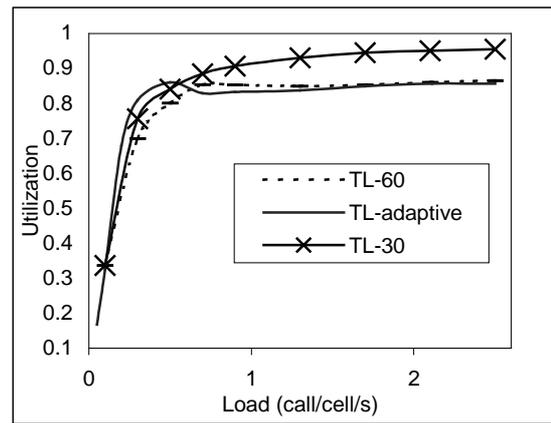


Figure 15. Utilization: Adaptive vs. Fixed.

VI. CONCLUDING REMARKS

In this paper, we propose a novel CAC scheme for wireless and mobile networks that avoids per-user reservation signaling. In order to predict the expected bandwidth of future handoffs we utilized two time series-based methods: an adaptive exponential smoothing method, called Trigg and Leach (TL), which is effortless and does not impose computation overhead on the network elements and, the ARIMA-based method that requires a training period for model selection. In addition, TL method does not require a huge amount of saved data to perform forecasting, but ARIMA-based does. Our approach can also grant an upper bound on the handoff dropping probability even under higher loads based on the choice of an adequate prediction interval. We also have proposed and evaluated an adaptive algorithm to dynamically adjust the TL’s prediction interval in order to optimize the bandwidth utilization depending on the HDP objectives.

REFERENCES

- [1] D. Hong, S. Rappaport, “Traffic Model and Performance Analysis for Cellular Mobile Radio Telephone Systems with Prioritised and Nonprioritised Handoff Procedures,” *IEEE Tran. Vehic. Tech.*, Aug. 1986.
- [2] S. Choi and K. G. Shin, “Predictive and Adaptive Bandwidth Reservation for Handoffs in QoS Sensitive Cellular Networks,” In *Proc. ACM SIGCOMM’98*.
- [3] M. Naghshineh and M. Schwartz, “Distributed Call Admission Control in Mobile/Wireless Networks,” *IEEE JSAC*, 14(4), May 1996. pp. 711-717.
- [4] M.H. Chiu and M. A. Bassiouni, “Predictive Schemes for Handoff Prioritization in Cellular Networks based on Mobile Positioning,” *IEEE JSAC*, 18(3), Mar. 2000
- [5] Z. Xu et al. “ A New Adaptive Channel Reservation Scheme for Handoff Calls in Wireless Cellular Networks,” *Proc. of IFIP Networking2002*. pp 672-684, 2002.

- [6] W.-S. Soh and H. S. Kim, "Dynamic Guard Bandwidth Scheme for Wireless Broadband Networks," Proc. IEEE Infocom'01, Anchorage, AK, Apr. 2001
- [7] W.-S. Soh and H. S. Kim, "QoS Provisioning in Cellular Networks Based on Mobility Prediction Techniques," IEEE Comm. Mag., Jan. 2003, pp 86-92.
- [8] T. Zhang et al. "Local Predictive Resource Reservation for Handoff in Multimedia Wireless IP Networks," IEEE JSAC, 19(10), Oct. 2001.
- [9] D. W. Trigg, D. H. Leach, Exponential Smoothing with an Adaptive Response Rate, Operational Research Quarterly, vol. 18, 1967, pp. 53-59
- [10] W. A. Fuller, Introduction to statistical time series (New York: John Wiley & Sons, Inc., 1996).
- [11] Andrew C. Harvey, "Time Series Models", Cambridge: MIT Press, 2nd. Ed., 1993
- [12] S F. L. Fernandes *et al.* "Time Series Applied to Network Traffic Prediction: A Revisited Approach," In International Conference on Applied Modelling and Simulation-AMS 2002, MA, USA.
- [13] L. Bow and W. Leroy, "Toward an all-IP based UMTS System Architecture," IEEE Network, 15(1), Jan-Feb 2001, pp.36-45.
- [14] R. Berezdivin et al., "Next-Generation Wireless Communications Concepts and Technologies," IEEE Comm. Mag., 40(3), Mar 2002, pp. 108-116.
- [15] R. Braden et al. "Integrated Services in the Internet Architecture: an Overview," IETF RFC 1633, Jun. 1994.
- [16] S. Blake et al. "An architecture for Differentiated Services," IETF RFC 2475, Dec. 98.
- [17] K. Nichols et al, "A Two-bit Differentiated Services Architecture for Internet. RFC 2638, Jul 1999.

Kelvin Lopes Dias received the B.S. degree in computer science from Federal University of Pará (UFPA), in 1994, and the M.Sc and Ph. D. degrees in computer science from Federal University of Pernambuco (UFPE), in 1998 and 2004, respectively. Currently, he is a research fellow at GPRT (Networking and Telecommunications Research Group) at Federal University of Pernambuco. His research interests include wireless and mobile networks, quality of service in all-IP networks and performance evaluation.

Stênio Flávio de L. Fernandes. In 1992 and 1996 he graduated as Electronic Engineer and M.Sc. degree, respectively, from the Universidade Federal da Paraíba - Campus II, Campina Grande-PB, Brazil. In 1994 he joined the Centro Federal de Educação Tecnológica de Alagoas (CEFET-AL) in Brazil. At the moment, he is a Ph.D. Candidate in the Universidade Federal de Pernambuco (UFPE). His research activities focus on Network Traffic Modeling, Performance Evaluation, Congestion Control and Quality of Service (QoS) in the Internet. He is also a member of the Telecommunications and Networking Research Group (GPRT)

Djamel Fawzi Hadj Sadok received a B.S. degree in Systems Engineering from L'Institut de mathématiques de L'Université des Sciences et de la Technologie D'Alger, in 1985 and a Ph.D. degree in Computing from University of Kent at Canterbury, United Kingdom, in 1990. He is currently a professor at the Universidade Federal de Pernambuco (UFPE) and his research activities include work on Quality of Service IP Networks, Voice over IP and Cellular Telephony. He is also the leader of the Telecommunications and Networking Research Group (GPRT)

Análise, Simulação e Correções do Protocolo MMA para Multicast em Redes Móveis Estruturadas

Renato Maia, Vagner Sacramento, Markus Endler

Departamento de Informática – Pontifícia Universidade Católica do Rio de Janeiro

Rio de Janeiro – RJ – Brazil

{maia,vagner,endler}@inf.puc-rio.br

Abstract - In this article we describe the prototyping and analysis by simulations of the MMA protocol (Multicast by Multicast Agent), which aimed at analyzing its behavior and evaluating its deficiencies and limitations related to the multicast message delivery for wireless mobile networks. Based on a rigorous analysis and tests in deterministic scenarios we identified that the protocol had serious problems. We made the necessary modifications to the protocol, and then compared the original and the modified versions in several stochastic scenarios. The results showed that our version has a better performance than the original MMA protocol

Resumo - Neste artigo foi realizada uma prototipação e análise por simulações do protocolo MMA (Multicast by Multicast Agent), com intuito de analisar o seu comportamento e avaliar suas deficiências e limitações quanto à entrega de mensagens multicast para redes móveis sem fio. A partir da análise rigorosa e de testes realizados em cenários de simulação determinísticos identificamos que este protocolo apresentava sérios problemas de funcionamento. Fizemos as modificações necessárias e em seguida comparamos a versão original e a modificada em vários cenários de simulação estocásticos. Os resultados mostraram que a nossa versão apresenta um desempenho melhor do que o protocolo MMA original

I. INTRODUÇÃO

Muitas aplicações distribuídas necessitam transmitir a mesma informação para múltiplos destinos, como por exemplo, aplicações de video-conferência, processamento distribuído, jogos, dentre outras. Uma maneira eficiente de enviar dados de uma fonte para múltiplos destinos é a utilização de um serviço de *multicast*. Alguns protocolos de *multicast* existente na rede fixa (rede cabeada) são: DVMRP [DEERING 88] e MOSPF [MOY 94].

Um serviço de *multicast* provê uma forma eficiente de comunicação multiponto entre vários *hosts*, pois possibilita mais rapidez na entrega das mensagens e principalmente o uso adequado dos recursos de rede (e.g. largura de banda). Sendo assim, este serviço é apropriado para diferentes tipos de rede, incluindo as redes móveis infra-estruturadas, onde os enlaces sem fio apresentam largura de banda reduzida e maior probabilidade de falhas. Além de implementar as funcionalidades requeridas para a disseminação eficiente das mensagens na parte cabeada da rede móvel, os protocolos de

Multicast para redes móveis devem também tratar a migração de *hosts* entre pontos de acesso da rede (*handoff*¹).

Existem várias propostas [Suh, Shin and Kwon 2001], [Sadok, Cordeiro and Kelner 2000], [Acharya 1996], [Acharya 2000] para protocolos multicast que atendam as exigências das redes móveis sem fio infra-estruturadas.

Um dos principais problemas a serem tratados pelos protocolos de Multicast para redes móveis sem fio é realizar a entrega de datagramas multicast pelo melhor caminho levando em conta a mobilidade dos *hosts*. Para comunicação unicast de datagramas o protocolo IP móvel [Perkins 1997] é uma solução que garante a transparência de mobilidade. Isto é feito através da utilização de um Home Agent (HA) para cada *host* móvel, que trata do encaminhamento de datagramas para a localização corrente deste *host*. Para cada *host* móvel que esteja fora da sua rede origem, o HA correspondente mantém em um cache a informação sobre a rede na qual este *host* se encontra no momento (rede visitada). Este endereço é denominado *care-of-address*. Todo datagrama destinado ao *host* móvel é primeiramente encaminhado para o seu HA, e quando este identifica que o *host* móvel (MH) está fora da rede de origem, o HA encapsula o datagrama original em um novo datagrama IP (tunelamento) e o re-encaminha para um Foreign Agent (FA), na atual rede visitada pelo *host* móvel. Este re-encaminhamento é realizado com base no *care-of-address*. Ao receber os datagramas, o FA desencapsula e re-encaminha os datagramas para o *host* móvel visitante.

Neste artigo, foi realizada uma prototipação e análise por simulações do protocolo MMA (Multicast by Multicast Agent) proposto por [Suh, Shin and Kwon 2001], com o objetivo de compreender melhor o seu funcionamento e avaliar a sua complexidade e eventuais limitações. Como resultado desta análise identificamos uma série de problemas na especificação deste protocolo, e propusemos e implementamos soluções para corrigir estes problemas, fazendo com que o protocolo possa de fato ser usado em uma rede móvel. Para a prototipação e simulação do protocolo MMA original e do protocolo modificado utilizamos o ambiente MobiCS [Rocha and Endler 2001].

O restante do artigo está estruturado da seguinte forma: a seção 2 apresenta alguns trabalhos relacionados; a seção 3 descreve resumidamente o protocolo MMA; a seção 4 descreve a nossa implementação do protocolo

¹ Operação realizada durante a migração de um *host* móvel de uma rede para outra.

MMA, apresentando brevemente o framework MobiCS e alguns cenários utilizados na simulação, onde são apresentados os problemas encontrados e as soluções propostas para o protocolo MMA; a seção 5 apresenta uma simulação estocástica do protocolo e a seção 6 contém a conclusão deste trabalho.

II. TRABALHOS RELACIONADOS

II.1. IP MÓVEL

A especificação do IP Móvel contempla duas abordagens para suportar serviços de *multicast* para os *hosts* móveis [Xylonmenos and Polyzos 1997]: *multicast* baseado no *foreign agent* e *multicast* baseado no *home agent* [Chikarmane and Williamson 1998]. Estas abordagens utilizam os mesmos princípios do IP móvel para prover a transparência de mobilidade de *hosts* para datagramas *multicast*.

No *multicast* baseado no *foreign agent*, o *host* móvel tem que se registrar no grupo *multicast* sempre que ele move para uma rede visitada. Este esquema tem a vantagem de oferecer um caminho de roteamento ótimo e a inexistência de datagramas duplicados, além de não ser necessário qualquer encapsulamento para encaminhamento da mensagem. Contudo, quando o *host* móvel apresenta alta taxa de mobilidade, o gerenciamento da árvore de *multicast* pode gerar muito overhead.

No *multicast* baseado no *home agent*, o datagrama *multicast* é encaminhado através do tunelamento *unicast* do IP móvel via HA. Quando o *home agent* recebe um datagrama *multicast* destinado para um *host* móvel, ele encapsula este datagrama duas vezes (uma vez com o endereço IP do *host* móvel, e outra com o care of address do FA) para então transmiti-lo para o *host* móvel como um datagrama *unicast*. O encapsulamento duplo aumenta o tamanho do pacote, e os datagramas geralmente acabam sendo enviados por um caminho que não seja ótimo, uma vez que a rota de encaminhamento sempre passa pelo HA. Além do mais, se vários *hosts* móveis que pertencem à mesma rede local estão na mesma rede visitada (rede externa), cópias duplicadas dos datagramas *multicast* chegarão ao destino. Assim, nenhuma das duas propostas do IP móvel se mostram como uma boa solução para implementar serviços de *multicast*.

II.2. MOM (MOBILE MULTICAST)

O MoM [Harrison 1997] propõe um protocolo *multicast* baseado no *home agent*, sendo este responsável pelo tunelamento de datagramas *multicast* para os *hosts* móveis. O MoM essencialmente define soluções para alguns dos problemas existentes no *multicast* baseado no *home agent* do IP móvel, discutido na seção anterior.

No esquema de *multicast* baseado no *home agent* apresentado na seção 2.1, um *home agent* repassa uma cópia separada do datagrama *multicast* para cada *host* móvel. Se todos *hosts* móveis que desejam receber o datagrama *multicast* estiverem na mesma rede visitada, n cópias duplicadas serão encaminhadas para o mesmo *foreign agent*. Para resolver este problema, no protocolo MoM, o *home agent* repassa uma única cópia do

datagrama *multicast* para cada rede visitada que contém algum *host* móvel do grupo *multicast*. Ao receber o datagrama *multicast*, o *foreign agent* o encaminha para os *hosts* móveis usando um link em nível de *multicast*. Este esquema reduz o número de datagramas *multicast* duplicados e, economiza a banda passante nos links wireless e wired.

Um outro problema resolvido pelo protocolo MoM é a convergência de múltiplos túneis de *home agents* diferentes estabelecidos com um mesmo *foreign agent*, mostrado na Figura 1. Quando múltiplos *home agents* têm *hosts* móveis na mesma rede visitada, uma cópia de cada datagrama é repassada para o mesmo *foreign agent* por cada *home agent*. Como solução para este problema, o MoM propõe que o *foreign agent* defina somente um único *home agent* como sendo o seu Designated Multicast Service Provider (DMSP) para um dado grupo *multicast*. Assim, o DMSP envia um único datagrama através do túnel para o *foreign agent*, enquanto os outros *home agents* que não são um DMSP ficam aguardando sua vez para transmitir. A solução proposta através do DMSP é mostrada na Figura 2.

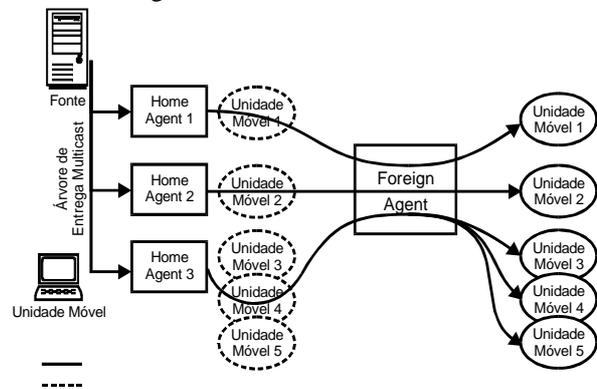


Figura 1. Problema de convergência de tunelamento.

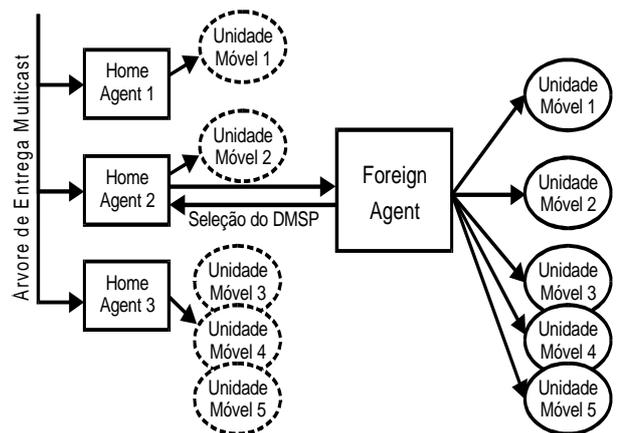


Figura 2. Seleção do DMSP no protocolo MoM.

Apesar de reduzir o tráfego *multicast*, o MoM apresenta problemas de duplicação de pacotes, e problemas na escolha de caminhos otimizados para entrega do datagrama *multicast* [Suh, Shin and Kwon 2001].

III. O PROTOCOLO MMA

O protocolo analisado, chamado de MMA (*Multicast by Multicast Agent*), propõe um novo esquema de

comunicação multicast, introduzindo os conceitos de *Multicast Agent (MA)* que vai estar presente em uma estação de suporte de mobilidade (*Mobile Support Station – MSS*) e *Multicast Forwarder (MF)* [Suh, Shin and Kwon 2001]. Essencialmente, os MAs são *foreign agents* que são responsáveis por repassar os datagramas multicast para os hosts móveis presentes na sua rede local. Um MF de um grupo multicast é um MA que além disto faz parte da árvore de multicast do grupo. Cada MA define um MF do qual deverá receber retransmissões de datagramas para os hosts móveis de um grupo *multicast*. Esta associação de um único MF por MA é definida dinamicamente, levando-se em consideração também a indicação de um MF por um host móvel a cada *handoff*. Quando a rede local de um MA faz parte da árvore de entrega *multicast*, o próprio MA pode fazer o papel de MF. Mas o MF associado a um MA cuja rede não faz parte da árvore *multicast* deverá ser um MA de outra rede (que pertença ao grupo *multicast*). Além disto, a associação de um MF para um MA é específica para cada grupo *multicast*.

No protocolo MMA, são utilizados dois métodos distintos para a definição da fonte de datagramas multicast, dependendo se a rede sendo visitada por um MH pertence ou não a uma árvore de multicast. Se a rede sendo visitada pertence à árvore multicast, os dados repassados ao host móvel são obtidos pelo MA diretamente do roteador multicast local da rede. Se a rede sendo visitada não pertence à árvore multicast, os dados multicast são obtidos através de um túnel do MF, ou seja, um MA que esteja incluso na árvore do grupo multicast e que esteja localizado em uma rede próxima da rede sendo visitada pelo host móvel.

III.1. DESCRIÇÃO DO PROTOCOLO

Essencialmente, o protocolo MMA funciona da seguinte maneira: quando um *host* móvel migra de uma rede (e.g. N1) para outra rede (e.g. N2), este envia a informação sobre o seu MF (MA da rede N1) para o MA na rede N2 durante o processo de registro na nova rede, informação esta que é utilizada pelo MA para selecionar o novo MF. Se N2 pertence à árvore de entrega *multicast*, o próprio MA de N2 se torna o MF, e tanto o MA quanto o *host* móvel atualizam os seus registros sobre o *MF corrente* com a identificação do MA de N2. Se o MA de N2 não pertence à árvore de entrega *multicast* e o MA não possui informação sobre MF no grupo *multicast*, o registro de *MF corrente* que do *host* móvel é usado também em N2. Se o MA em N2 não pertence a árvore de entrega *multicast*, mas o MA tem informação sobre um MF no grupo *multicast*, há duas possibilidades: a) Na primeira o valor de MF que o *host* móvel possuía em N1 é usado diretamente como MF em N2 (seleção do MF mais antigo), ou b) Na segunda, o MA seleciona o agente que está mais próximo a ele dentre aquele informado pelo *host* móvel e aquele anteriormente conhecido pelo MA (seleção do MF mais próximo). Em ambos os casos, o MA e o *host* móvel são atualizados com o valor do MF escolhido.

O registro na árvore multicast deve ser requisitado pelos hosts móveis visitantes durante o processo de registro em um determinado grupo multicast. Sempre que

um *host* móvel migra para uma nova rede e se registra com um novo MA, o novo MA executa o processo de registro à árvore multicast caso tal funcionalidade tenha sido requisitado pelo *host* móvel. Este processo é similar ao dos protocolos multicast baseados em *foreign agents* (conforme descrito na seção 2.1).

Se o *host* móvel se “conecta” a um MA que não faz parte da árvore multicast, este MA estabelecerá um túnel com o MF para que este retransmita as mensagens encaminhadas para o grupo multicast do qual o *host* móvel faz parte. A confirmação do estabelecimento deste túnel no MA da rede visitada ocorre através do recebimento do primeiro datagrama multicast do grupo dentro de um intervalo de tempo pré-estabelecido.

IV. IMPLEMENTAÇÃO

Nesta seção apresentaremos o ambiente utilizado para a prototipação e simulação do protocolo MMA, e explicaremos algumas simplificações sobre a rede que fizemos em nossas simulações. Em seguida, discutiremos os principais problemas detectados na especificação original do protocolo MMA (contida em [Suh, Shin and Kwon 2001]) a partir de cenários específicos de execução, e finalmente apresentaremos as nossas soluções para os problemas, que foram implementadas em uma versão modificada do MMA.

IV.1. MOBICS

MobiCS [Rocha and Endler 2001] é um *framework* desenvolvido em Java que permite a criação de um ambiente customizado para a prototipação e simulação de protocolos distribuídos para redes móveis infra-estruturadas. O *framework* permite que usuário defina as propriedades dos elementos de rede (hosts móveis, estações fixas, pontos de acesso, enlaces, etc.) e a configuração da rede. Além disto, induz o usuário a programar o seu protocolo de forma modular e totalmente independente do(s) modelo(s) de simulação usado.

Um modelo de simulação define todos os efeitos externos ao protocolo sendo prototipado, tais como o padrão de mobilidade dos hosts móveis, a relação de vizinhança entre as redes sem fio, o padrão de conectividade dos hosts móveis e propriedades dos enlaces sem fio, entre outros. MobiCS permite tanto o uso de modelos pré-concebidos, como também a criação de novos modelos de simulação, dando assim ao usuário a liberdade de criar um ambiente de simulação com um grau variável de detalhes e com abstrações adequadas para avaliar certas propriedades de seu protocolo. Por exemplo, para alguns protocolos pode ser necessário definir e programar o roteamento de mensagens/pacotes entre hosts fixos, enquanto que em outros protocolos isto pode ser irrelevante, bastando que se defina uma latência média para comunicação entre quaisquer dois hosts fixos.

Em MobiCS, são previstos dois modos de simulação: o determinístico e o estocástico. No modo determinístico o usuário define um (ou mais) script de simulação, cada qual descrevendo uma situação (ou cenário) bem específica, na qual deseja testar o seu protocolo. Neste script, são indicados explicitamente eventos como

migrações de hosts, desconexões e reconexões, bem como as requisições do serviço implementado pelo protocolo sendo testado (por exemplo, uma entrega multicast) provenientes de uma suposta “aplicação cliente”. Além disto, define-se nestes scripts a ordem relativa (causal) de ocorrência destes eventos. Simulando o protocolo segundo cada um destes scripts, o usuário é capaz de detectar se o seu protocolo apresenta o comportamento esperado para o cenário correspondente. Portanto, o modo determinístico é geralmente utilizado para a depuração de protocolos.

No modo estocástico o objetivo é testar o comportamento e avaliar a complexidade de um protocolo em cenários maiores e semi-aleatórios. Neste modo, a geração dos eventos acima citados é determinada por probabilidades escolhidas pelo usuário. Para isto, o usuário programa geradores de eventos aleatórios para cada tipo de evento (migração, desconexão, atraso, falha, etc.). Por exemplo, para um host móvel, os eventos relevantes são geralmente migrações, desconexões e reconexões. Em ambos os modos, o usuário é capaz de instrumentar o seu simulador a fim de poder contar número de ocorrências de eventos, ou visualizar o estado das estruturas de dados usadas em seu protocolo.

IV.2. SIMPLIFICAÇÕES ADOTADAS

Ao invés de simular o MMA para uma configuração de rede complexa e possivelmente com centenas de parâmetros, adotamos algumas simplificações a fim de manter a simulação pequena e gerenciável. No entanto, as simplificações adotadas foram definidas de forma a não influenciar ou induzir um determinado comportamento do protocolo. Nesta sub-seção são descritas algumas simplificações adotadas para a simulação do protocolo MMA no ambiente MobiCS. Apesar destas simplificações, objetivou-se a conformidade total com à especificação do protocolo original. As simplificações adotadas foram definidas de forma a não alterar o princípio de funcionamento do protocolo, estas são descritas abaixo:

Deteção de migração dos hosts móveis: A descrição original do protocolo sugere que a migração dos hosts móveis seja detectada através de um time-out na rede de origem. Dessa forma é necessário que os hosts móveis enviem periodicamente um beacon (ou uma mensagem IamAlive) para o atual MA informando que ainda se encontram (ativos) na rede deste MA. Entretanto esta abordagem se mostrou difícil de ser implementada no MobiCS. Por esta razão, abstraiu-se desta deteção, e em vez disto, implementou-se que no momento da migração para a nova rede, o host móvel informa ao novo MA a rede da qual ele está vindo. Assim, o novo MA pode enviar uma mensagem ao antigo MA informando-o da migração do host.

Seleção do ótimo MF: A descrição do protocolo MMA não define um critério específico para selecionar o MF mais próximo/adequado no momento da migração de um host móvel que traga consigo o registro de um MF. Na implementação apresentada neste trabalho é feita a seleção do último MF (oldest selection), que consiste em sempre escolher o MF trazido pelo host móvel.

Topologia da rede cabeada: Todos os cenários avaliados são baseados numa rede composta por um conjunto de estações de suporte à mobilidade (MSSs, ou MAs, na terminologia do MMA) interligadas pelo meio cabeado (wired). Dentre as quais existe um subconjunto dessas estações que estão ligadas com um roteador multicast (MR), podendo portanto fazer parte de alguma árvore de entrega multicast. O roteador multicast é o núcleo da rede e é responsável por receber e repassar as mensagens de multicast para os MAs conforme ilustrado na Figura 3.

MHs estacionários: Para evitar que alguns MAs saiam da árvore de multicast em virtude da migração de todos os seus MHs, em alguns casos um MH é mantido estacionário em cada MA, de forma que este nunca saia da árvore de multicast.

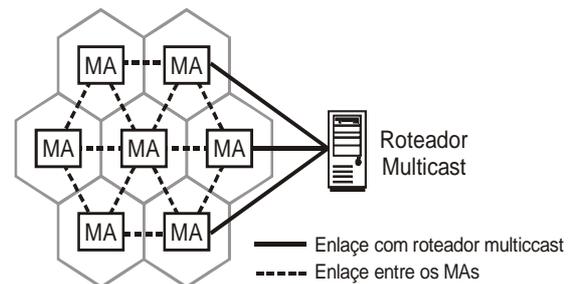


Figura 3. Conjunto de MAs interligados a um roteador multicast.

IV.3. CENÁRIOS DETERMINÍSTICOS

Nesta seção, são apresentados os cenários determinísticos que permitiram a deteção dos problemas do protocolo MMA. Antes de programar o *script* que descreva o cenário específico a ser simulado no MobiCS, deve-se determinar qual é exatamente o comportamento do protocolo a ser testado em cada cenário, escolher o conjunto mínimo de elementos de rede e definir a ordem (causal) da ocorrência de eventos externos ao protocolo que sejam necessários para criar a situação desejada. Em seguida, a sequência de eventos são codificadas no *script*. O MobiCS provê várias primitivas para a programação do *script*, tais como, pontos de sincronização global, ocorrência de uma migração ou de uma desconexão/reconexão de um MH, habilitar/desabilitar a recepção de mensagens ou eventos de *timer*, dentre outros.

O MMA foi simulado em vários cenários diferentes com o intuito de testar o seu comportamento em várias situações extremas ou possivelmente não previstas pelos seus autores, como por exemplo, quando mensagens são perdidas ou recebidas fora da ordem esperada. Nestas simulações avaliamos a eficiência e a corretude do protocolo, por exemplo, se ocorre duplicação e/ou perda de mensagens no serviço de multicast.

IV.3.1. PROBLEMAS DO PROTOCOLO ORIGINAL

CENÁRIO 1 - PROBLEMA DA SAÍDA DEFINITIVA DO MA DA ÁRVORE DE MULTICAST

O protocolo MMA determina que quando todos os MHs de um determinado MA migram para uma outra

rede sem fio (célula) e, este MA pára de repassar mensagens multicast para outros MAs por tunelamento, então ele deve sair da árvore de multicast através do envio de uma mensagem PRUNE. Esta característica pode levar a diversas situações indesejáveis, como por exemplo, que um dado MA saia da árvore de roteamento multicast (devido ao recebimento de uma mensagem ForwardSTOP) e em seguida receba um pedido (atrasado) de tunelamento de um outro MA para um MH que o tenha indicado como MF. Esta situação é ilustrada no diagrama espaço-tempo da Figura 4, onde, após a migração do MH0 da rede N2 para N3, o MA N1 sai da árvore

multicast (de-registrando-se junto ao MR) pois supostamente não possui mais qualquer MH do grupo sob sua responsabilidade, e além disto recebe a mensagem ForwardSTOP antes da mensagem ForwardREQUEST.

Note que este problema impossibilita a utilização do protocolo, pois não há meios de se garantir que um dado MA tenha sempre algum MH membro do grupo em sua rede, ou então mantenha sempre a função de MF para MHS em outras redes. Em sua versão original, portanto, o protocolo MMA possibilitaria que o serviço multicast se tornasse indisponível de forma permanente em vários pontos de acesso.

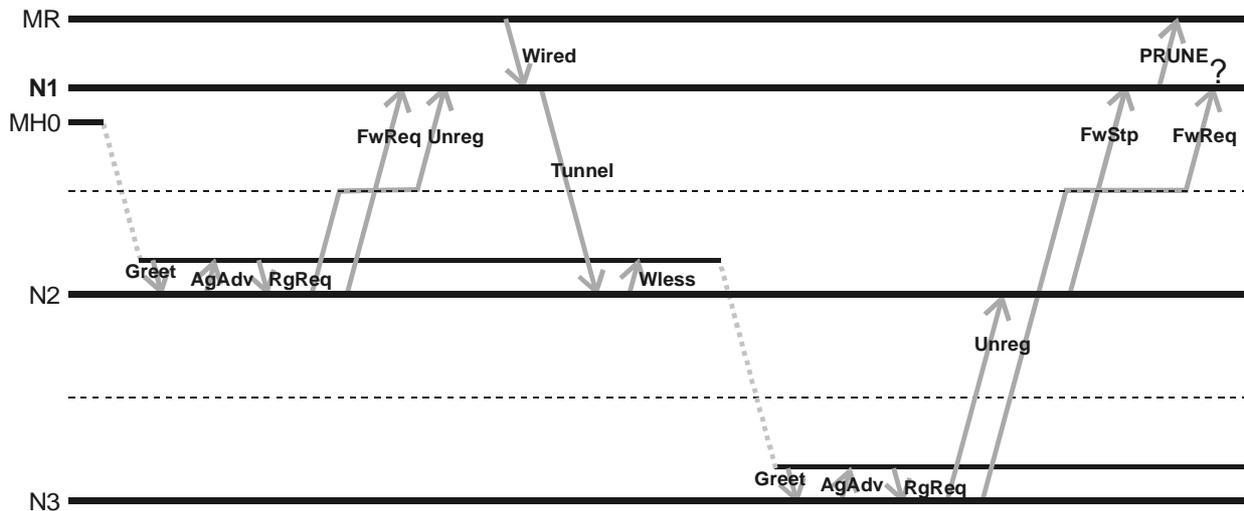


Figura 4. Cenário mostrando o problema da saída prematura do MA da árvore multicast.

CENÁRIO 2 – PROBLEMA DOS MÚLTIPLOS TÚNEIS ABERTOS

A especificação do protocolo MMA define que o estabelecimento de túneis entre os MAs seja feito pelo envio de mensagens de controle (i.e. ForwardREQUEST) que podem ser perdidas caso não haja rota disponível entre os MAs. Desta forma, é necessário algum mecanismo para a confirmação do estabelecimento do novo túnel, a fim de permitir que o túnel anterior/antigo seja encerrado. Essa confirmação ocorre no momento do recebimento do primeiro datagrama tunelado proveniente do novo MF antes de um determinado tempo de espera (i.e. timeout). Entretanto, a especificação do protocolo MMA não trata a situação na qual este primeiro datagrama enviado pelo novo túnel chega após o tempo de espera. Assim sendo, o protocolo MMA permite que vários túneis para um mesmo MH em um grupo sejam mantidos abertos simultaneamente, o que acarreta um problema crítico de duplicação de mensagens.

Esta situação é ilustrada pelo cenário da Figura 5, onde o MA da rede N3 inicialmente está recebendo datagramas tunelados pelo MA da rede N1 (supondo-se que N1 tenha sido supostamente indicado como MF de MH1 (Fases 1 e 2)). Entretanto, quando o MH0 (originalmente sitiado na rede N1) migrar da rede N2 para a rede N3 (Fase 5), um novo pedido de tunelamento será enviado para o MA de N2. Sendo assim, o MA da rede N3 terá associado a si temporariamente dois túneis para o mesmo grupo multicast. O túnel estabelecido com N1 só será finalizado quando a confirmação de estabelecimento

de tunelamento de N2 for recebida. De acordo com o cenário ilustrado, o MA de N3 receberá mensagens duplicadas (Fase 6), uma enviada pelo MA de N1 e outra enviada pelo MA de N2, e ambas serão repassadas para todos MH's locais que fazem parte do grupo multicast.

Essa situação tem uma agravante, pois um número arbitrário de mensagens duplicadas pode ser continuamente recebidas e repassadas para os MH's locais. A duplicação de datagramas ocorrerá também sempre que a confirmação de estabelecimento do tunelamento enviada por N2 chegar depois que o timeout ocorrer, pois o protocolo MMA não trata adequadamente este caso e, portanto, nunca enviará espontaneamente a mensagem ForwardStop para N1. Ou seja, N3 continuará recebendo datagramas multicast através de dois túneis indefinidamente. Novamente, o problema está no fato de que a especificação do protocolo MMA não define um procedimento adequado para encerrar os túneis abertos.

Este problema faz com que o protocolo MMA não satisfaça a propriedade essencial do protocolo multicast, que é o de evitar que a mesma mensagem seja transmitida a partir de várias fontes diferentes para o mesmo destino. Além do mais, há um desperdício da banda passante na rede cabeada, e também no meio sem fio devido à retransmissão de mensagens duplicadas para os hosts móveis.

CENÁRIO 3 – PROBLEMA DO RETORNO DO MH

Quando um MH migra para uma nova rede, durante o processo de registro do MH junto ao novo MA é

selecionado o melhor MF dentre o atual utilizado pelo novo MA e aquele informado pelo MH. O protocolo MMA verifica se o MF selecionado é igual ao MF informado pelo MH, e neste caso é capaz de deduzir o seguinte: (a) o MA está recebendo as mensagens de multicast por um túnel com outro MA, pois se o MA estivesse incluso na árvore de multicast, o MF escolhido

seria o próprio MA, que sempre seria selecionado como MF ótimo por possuir um roteador multicast disponível. (b) um novo túnel deve ser estabelecido com o MF informado pelo MH, pois este foi o MF escolhido. Com base nestas informações o protocolo MMA decide estabelecer um novo túnel com o MF escolhido.

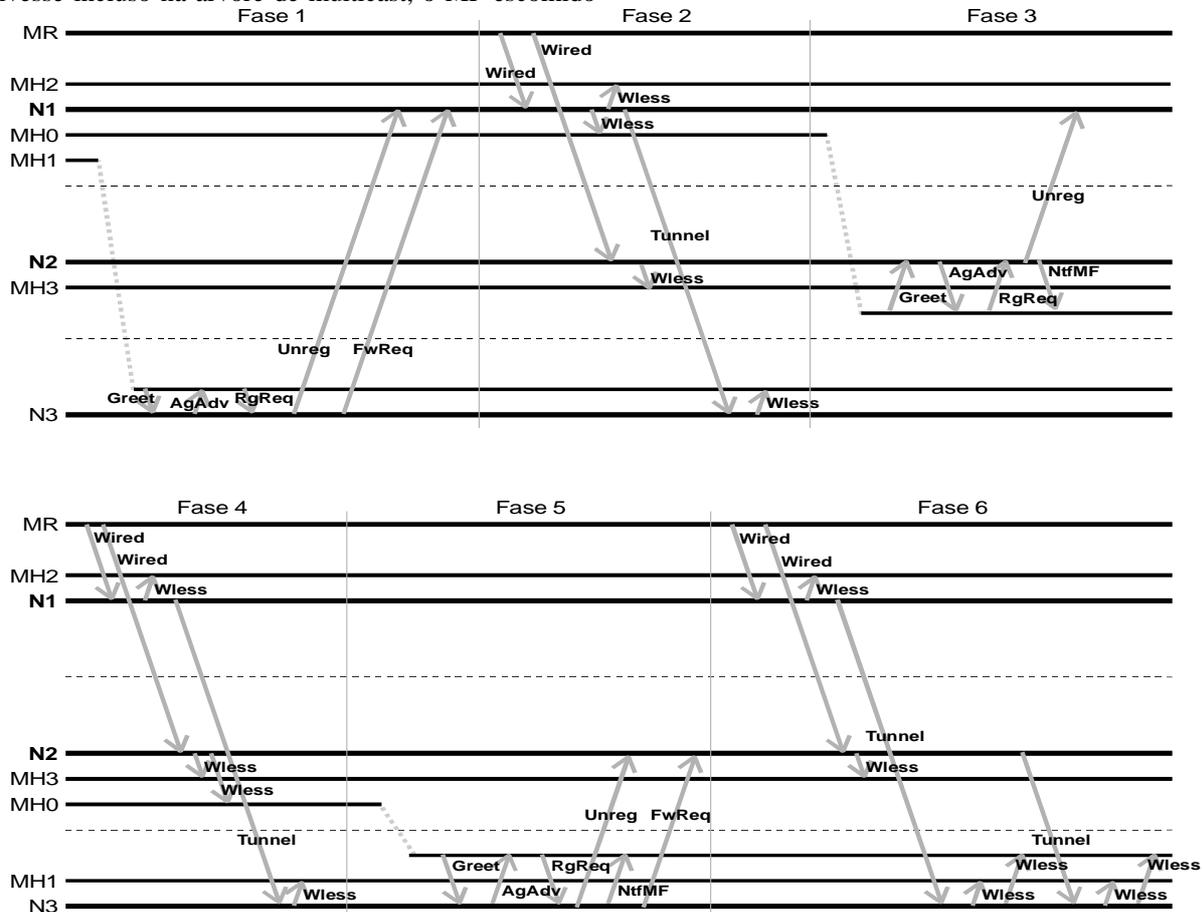


Figura 5. Cenário mostrando o problema da abertura de múltiplos túneis.

É importante notar que as deduções discutidas anteriormente somente podem ser feitas corretamente se o MF informado pelo MH for diferente do MF utilizado pelo MA. Entretanto isso nem sempre ocorre, pois mesmo após migrar para outras redes o MH ainda pode manter o mesmo MF, pois este pode também ser selecionado como MF ótimo. Desta forma, é possível que o MH retorne a uma rede trazendo consigo o mesmo MF associado ao MA. Neste caso, podem ocorrer duas situações: (a) se o MA estiver recebendo datagramas através de um túnel com outro MA, então será feita uma solicitação para o estabelecimento de um novo túnel com o MF atual. Entretanto, o MA aguardará a confirmação do

estabelecimento do “novo” túnel. Se um datagrama multicast for enviado antes do tempo para confirmação o novo túnel, o túnel atual será encerrado. Note que neste caso o serviço de entrega multicast se torna indisponível até que o MA decida trocar de MF. (b) se o MA fizer parte da árvore, então será feita uma solicitação para o estabelecimento de um túnel com o próprio MF. A formação de túneis em círculo é extremamente desastrosa, pois o recebimento de um novo datagrama faz com que o MA fique bloqueado indefinidamente num loop de envio de mensagens tanto pelo meio cabeado (através de túneis com outros MAs) quanto pelo meio sem fio, como ilustrado na Figura 6.

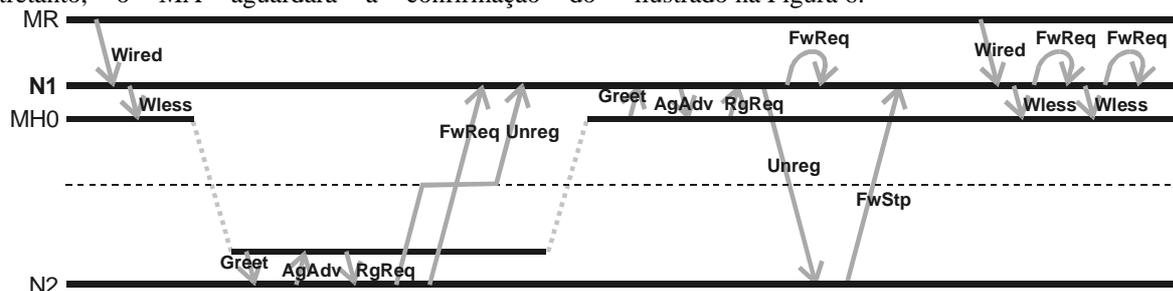


Figura 6. Cenário mostrando o retorno do MH utilizando o mesmo MF.

IV.3.2. O PROTOCOLO MMA MODIFICADO

Nesta seção apresentaremos algumas correções e extensões do protocolo que solucionam os problemas discutidos na seção anterior. As alterações foram incorporadas preservando a estrutura do protocolo e mantendo a essência de sua funcionalidade para comunicação multicast para redes móveis.

O problema apresentado no Cenário 1 se deve ao fato do protocolo MMA não tratar a situação de um MH eventualmente precisar usar um MA como MF que já tenha saído da árvore de multicast. É importante ressaltar que é correto fazer com que um MA saia da árvore de multicast caso não existam mais MH's em sua rede ou túneis associados ao grupo multicast, pois isso evita que datagramas multicast sejam enviados pelo meio cabeado desnecessariamente. Deve-se observar também que o quanto maior for o número total de MHs no sistema, menor será a probabilidade de um MA sair da árvore de multicast.

A solução proposta para a correção do problema consiste em fazer com que o MA se reintegre a árvore multicast caso receba um novo pedido de tunelamento de outro MA. Apesar da operação de reintegração à árvore ser computacionalmente custosa, essa abordagem se parece ser mais eficiente do que manter um MA na árvore multicast mesmo que para efeito da entrega multicast isto não seja necessário. Além do mais, acreditamos que o Cenário 1 seja relativamente raro, de forma que haverá poucos casos em que um MA saia e tenha que se reintegrar à árvore de multicast em um curto espaço de tempo. A implementação desta solução é mostrada no trecho de pseudo-código do algoritmo MMA para um MA (no papel de MF), ilustrado na Figura 7.

```
IF(Forwarding STOP message from a foreign MA is received){
  Delete the sender MA from FA_List from
  the group entry;
  IF(Group member does not exist in group
  entry anymore){
    Sends PRUNE message;}
}ELSE IF(Forwarding REQUEST message from a foreign MA
is received){
IF(Group entry does not exist in Group Information Table)
{
  Make new group entry;
  Send join message to multicast delivery tree;
}
  Add the sender MA to FA_List in group
  entry;
}
```

Figura 7. Correção do problema da saída prematura do MA da árvore de multicast.

O problema descrito no cenário 2 trata do estabelecimento de múltiplos túneis, pois o protocolo MMA não trata a possibilidade da confirmação do estabelecimento do túnel não ser recebida no tempo esperado (i.e. timeout). Sendo assim, datagramas multicast serão recebidos por túneis com MFs diferentes do MF corrente. Portanto, se um datagrama for recebido nessa circunstância, induz-se que este datagrama veio através de um túnel antigo, que então deve ser encerrado. Desta forma, elimina-se o problema de datagramas

duplicado nas mensagens subsequentes transmitidas para um dado grupo multicast.

A solução proposta é descrita no trecho de pseudo-código do algoritmo MMA mostrado na Figura 8, que demonstra o procedimento realizado quando um datagrama multicast é recebido.

O problema descrito no cenário 3 é devido o protocolo MMA não tratar o fato de um MH informar o mesmo MF utilizado pelo MA. Ao receber um novo pedido de registro de um MH não é necessário selecionar um novo MF se o informado pelo MH for o mesmo utilizado pelo MA.

```
IF(datagram is forwarded from the MF before time-out
occurs){
  Send Forwarding STOP message to old MF;
  Notify new MF to all mobile hosts;
  Update its MF with new MF;
}ELSE{
  Send Forwarding STOP message to the MF;
  Ignore tunnelled datagram;
}
```

Figura 8. Correção dos problemas de múltiplos túneis abertos.

```
Registration Request Message is received from MH
Take the information (Group_ID, MF, JOIN_Option);
IF(Group ID is already registered in Group Information
Table){
  Add MH to MH_List;
IF(MF received from MH different to MF){
  Inform MH of its MF value;
  Select new optimal MF;
  IF(Selected MF equals to MF received from MH){
    Send Forwarding REQUEST message to new MF;
    IF(datagram is forwarded from the MF before
time-out occurs){
      Send Forwarding STOP message to old MF;
      Notify new MF to all mobile hosts;
      Update its MF with new MF;
    }
  }
}
* On receiving a registration request.
Registration Request Message is received from MH
Take the information (Group_ID, MF, JOIN_Option);
IF(Group ID is already registered in Group Information
Table){
  Add MH to MH_List;
  IF (MF received from MH is not MA itself) {
    Inform MH of its MF value;
    Select new optimal MF;
    IF(Selected MF equals to MF received from MH)
    Send Forwarding REQUEST message to new MF;
  }
}ELSE{
  Make new group entry;
  Add MH to MH_List;
IF (MF received from MH is not MA itself) {
  Send Forwarding REQUEST message to MF of MH;
  Set up MF value with MF of MH;
} ELSE
  Send join message for connection to multicast
delivery tree;
}
```

Figura 9. Correção para os problemas de processamento de requisições para estabelecimento de tunelamento

Para resolver o problema descrito no cenário 3, é realizada uma validação do MF informado pelo MH. Se o MF especificado pelo MH na mensagem RegistrationRequest for o MF corrente utilizado pelo MA, as seguintes condições devem ser analisadas: Se o MA está na árvore multicast, o MH deve somente ser adicionado na lista de MHs, pelo contrário, o MA deve enviar uma mensagem Join para o MR, com o propósito de juntar-se a árvore para receber as mensagens multicast destinadas ao MH. Deve ser notado que nesta ocasião o MA nunca envia uma mensagem de ForwardRequest para o seu próprio endereço, conforme o problema abordado no cenário 3. Ao invés disso, outros procedimentos são tomados, dependendo se o MA está ou não na árvore multicast. Ressalta-se que estes procedimentos serão executados somente na ocasião em que um determinado MH, sair da rede que o MA se encontra, e em um momento posterior retornar para esta mesma rede tendo o MA como seu MF. É muito provável que os MHs troquem de MF durante o seu deslocamento pelas outras redes. A solução proposta resolve o problema abordado e não aumenta o custo de gerenciamento da árvore de multicast, pois ela vai ser executada somente em situações muito específicas. O código da Figura 9 mostra a implementação desta solução.

V. TESTES ESTOCÁSTICOS

Nesta seção é feita uma avaliação do impacto das alterações feitas no protocolo MMA sobre a sua performance através de simulações estocásticas. Nestas, foram feitas comparações do número de datagramas perdidos e duplicados no protocolo original e no protocolo modificado. Como o problema ilustrado no Cenário 3 permite a formação de um *loop* (MA tunela datagrama para si mesmo) para a simulação alteramos o protocolo original a fim de evitar este problema.

Nas simulações, criou-se uma infraestrutura composta por 19 redes (células) hexagonais, e onde todos os MAs das redes possuem uma ligação com o roteador multicast. Entretanto, apenas 10 redes selecionadas aleatoriamente compõem inicialmente a árvore multicast. Foram feitas simulações com 150 e 300 hosts móveis distribuídos uniformemente entre as redes que compõem a árvore multicast inicial. Não foi utilizado o Join Option, pois as modificações feitas no MMA não abrangem o tratamento deste recurso. De fato, a utilização do Join Option ameniza os problemas do protocolo original, uma vez que permite que um MA que tenha saído da árvore (Cenário 1) possa se reintegrar à mesma.

Nas simulações usou-se quatro taxas de mobilidade: muito baixa, baixa, média-baixa e média, isto é, a probabilidade de um host móvel migrar a cada 100 Unidades de Tempo Simuladas (UTS) é de 10%, 20%, 40% e 50%, respectivamente. Em cada simulação 200 datagramas são enviados em multicast para todos os hosts móveis, sendo que a probabilidade de um datagrama ser enviado (também a cada 100 UTS) é de 50%. O timeout utilizado para a confirmação do estabelecimento de um túnel foi de 200 UTS, fazendo com que a probabilidade do pedido de estabelecimento de um túnel seja confirmado no tempo esperado seja de 75%.

Para cada conjunto de parâmetros escolhido foram feitas 30 simulações. Os resultados apresentados para o MMA original e o modificado (corrigido) são a média aritmética dos valores medidos e as barras verticais indicam o desvio padrão. Foram medidos o número total de datagramas enviados por tunelamento, o número de datagramas perdidos, (i.e. a soma do número total de datagramas que não foram recebidos por cada host móvel) e o número de datagramas repetidos enviados pelo meio cabeado e pelo meio sem fio.

O gráfico da Figura 10 mostra o número de datagramas não recebidos pelos hosts móveis. Apesar do problema ilustrado pelo Cenário 1 indicar que pode ocorrer uma perda de um grande número de datagramas, este cenário é bastante raro, especialmente devido ao fato de que o protocolo original permite que um número maior do que necessário de túneis permaneçam abertos (Cenário 2), tornando menos provável que algum MA tenha sua entrada de grupo sem nenhum membro cadastrado.

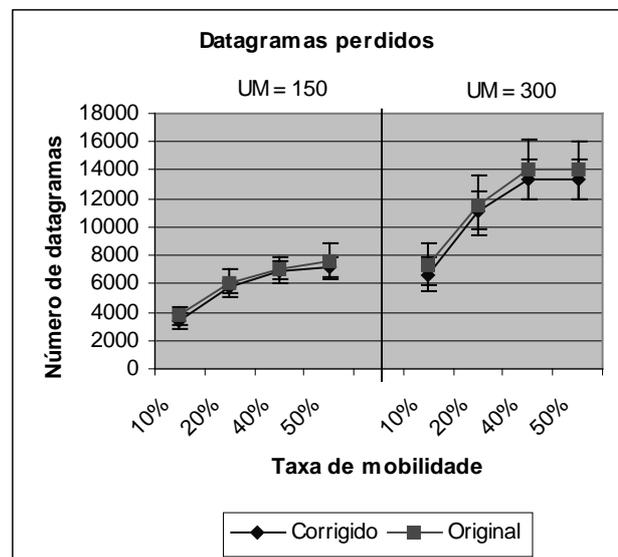


Figura 10. Número de datagramas não recebidos pelos hosts móveis em simulações feitas com 150 e 300 hosts móveis.

Nas simulações feitas, a saída de MA's da árvore multicast ocorre muito raramente e apenas com poucos hosts móveis. O fato do número de pacotes perdidos no MMA modificado ser apenas um pouco menor do que no MMA original, (apesar do problema discutido no Cenário 3) pode ser explicado da seguinte maneira, como no MMA original são deixados abertos desnecessariamente vários túneis (conforme mostrado no Cenário 2), esta redundância de envio de datagramas naturalmente reduz as chances de perda de datagramas, apesar do problema mencionado acima.

A Figura 11 mostra o gráfico do número de datagramas tunelados repetidos provenientes de diferentes túneis. Apesar da diferença entre o número de pacotes ser considerável, também neste caso o problema de duplicação de datagramas devido a múltiplos túneis (Cenário 2) pode estar sendo amenizado pelo fato de que a migração de um MH, que traz como MF um MA que mantém um túnel duplicado, pode fazer com que esse túnel seja utilizado corretamente e posteriormente seja encerrado de forma adequada.

O problema de replicação de datagramas se mostra ainda mais grave no meio sem fio, pois levando-se em consideração que cada datagrama redundante recebido por um MA resultará em vários datagramas duplicados sendo transmitidos pelo meio sem fio aos hosts registrados no MA. O gráfico da Figura 12 mostra o número de datagramas duplicados enviados pelo meio sem fio.

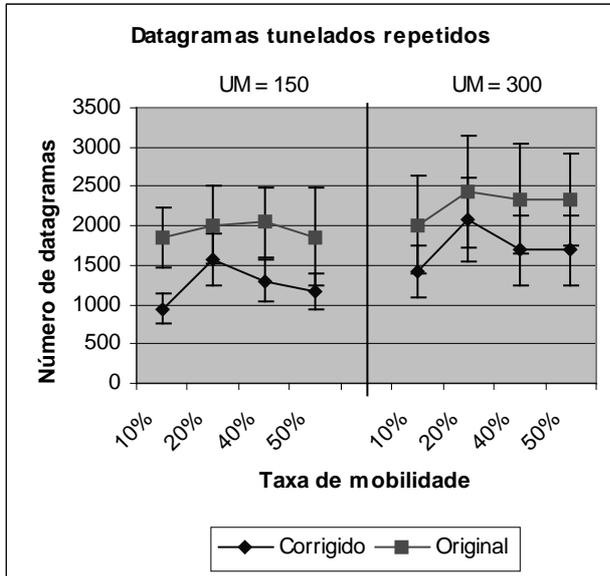


Figura 11. Número de datagramas repetidos recebidos pelos MA.

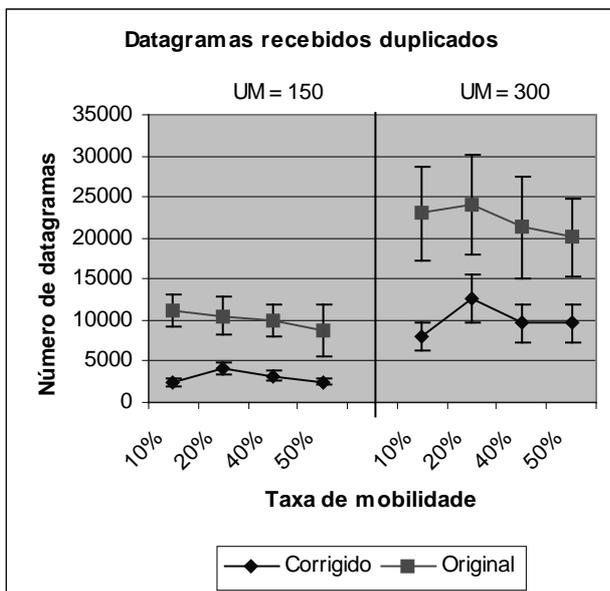


Figura 12. Número de datagramas repetidos recebidos pelos hosts móveis.

É interessante notar que os efeitos dos problemas do MMA original se amenizam mutuamente, pois o grande número de datagramas repetidos faz com que menos datagramas sejam perdidos e as situações que ocasionam a perda de datagramas permitem que menos datagramas repetidos sejam entregues. Por outro lado, isto não torna os problemas do protocolo original menos críticos, especialmente se levarmos em consideração a maior sobrecarga de comunicação (e processamento) causada por mensagens de controle trocadas entre os MAs para o

estabelecimento e encerramento de túneis (como pode ser visto no gráfico da Figura 13).

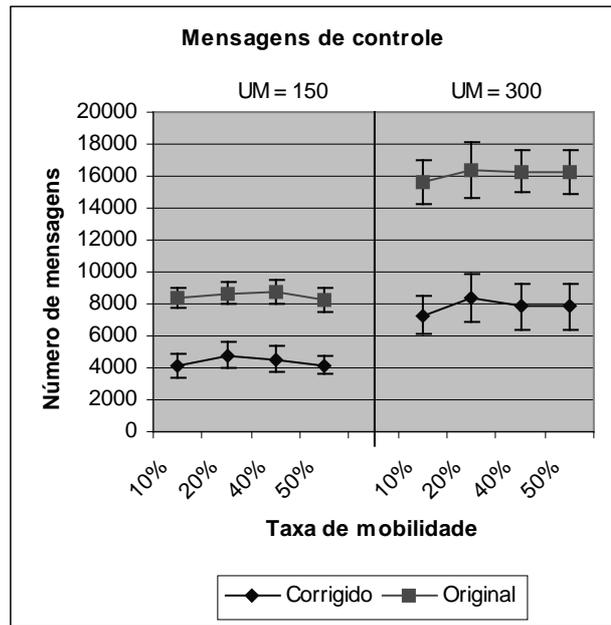


Figura 13. Número de mensagens trocadas entre os MAs para o estabelecimento e encerramento de túneis

VI. CONCLUSÕES

O desenvolvimento de protocolos para redes móveis é uma tarefa bastante complexa, para a qual são necessárias ferramentas para a especificação, o projeto, a prototipação, a simulação, a depuração e análise de desempenho destes protocolos. Devido à grande quantidade de variáveis que influenciam o comportamento de um protocolo em uma rede móvel, simulação tem sido uma abordagem amplamente utilizada para estes tipos de rede.

Neste artigo apresentamos o protocolo MMA para multicast em redes móveis, e discutimos alguns dos problemas deste protocolo para prover multicast de forma eficiente e correta, tais como, por exemplo, a duplicação de túneis, loops e o possível não atendimento dos hosts móveis. Para estes problemas propusemos pequenas modificações no protocolo que foram validadas através de uma prototipação e simulações do protocolo. Além disto, usamos simulações para comparar o MMA original e o modificado, e conseguimos mostrar vantagens do MMA modificado com relação à eficiência do serviço de multicast.

Entretanto, pudemos comprovar o quanto ferramentas para prototipação e simulação de protocolos, tais como por exemplo o MobiCS, podem ser úteis também para estudar e analisar protocolos já existentes ou publicados (como o MMA, em um journal conceituado) e possivelmente até corrigir ou otimizá-los. Mas estamos cientes também de que devido à necessidade de escolha de um determinado modelo de simulação (com certas simplificações/suposições) naturalmente, nunca será possível analisar todos os aspectos de um protocolo com igual ênfase. No nosso caso, o objetivo do trabalho foi analisar o aspecto de gerenciamento de mobilidade do

MMA, e acreditamos que os resultados obtidos foram bastante interessantes.

Como trabalhos futuros pretendemos comparar o MMA modificado com outros protocolos para multicast, como o MoM, e o protocolo proposto em [Sadok, Cordeiro and Kelner 2000].

VII. REFERÊNCIAS BIBLIOGRÁFICAS

- [Acharya 1996] Acharya A., Bakre A. and Badrinath B., IP multicast extensions for mobile internetworking, in: *IEEE INFOCOM'96* (1996).
- [Acharya 2000], Acharya A. and Badrinath B., A framework for delivering multicast message in networks with mobile hosts, *Mobile Networks and Applications* (2000) 199–219.
- [Bhatti 1999] Bhatti, N. and Schlichting. Configurable Communication Protocols for Mobile Computing, Proceedings of the 4th International Symposium on Autonomous Decentralized Systems, pages 220-227, Tokyo, March 1999
- [Chikarmane and Williamson 1998] Chikarmane, V. and Williamson, C.L.; Multicast support for mobile host using mobile IP: design issues and proposed architecture, *Mobile Networks and Applications* (1998) 365–379.
- [DEERING 88] DEERING, S. E. Multicast Routing in Internetworks and Extended LANs. Proceedings of ACM SIGCOMM'88 Conference, 1988, pp55-64.
- [Harrison 1997] Harrison T., Williamson C., W. Mackrell and R. Bunt, Mobile multicast (MOM) protocol: multicast support for mobile hosts, in: *Proc. Of ACM MOBICOM'97* (1997) pp. 151–160.
- [MOY 1994] MOY, J. Multicast Extensions to OSPF, Request For Comments 1584, Mar 1994, 102pp.
- [Perkins, 1997] Perkins, C. "Mobile IP," *IEEE Comm.*, Vol. 35, No. 5, 1997, pp. 84-99.
- [Rocha and Endler 2001] Rocha, Ricardo C.A. and Endler Markus. MobiCS: An Environment for Prototyping and Simulating Distributed Protocols for Mobile Networks. In Proc. 3rd IEEE Intern. Conference on Mobile and Wireless Communications Networks (MWCN2001).
- [Sadok, Cordeiro and Kelner 2000] Sadok Djamel H., Cordeiro Carlos de M. and Kelner Judith; *A Reliable Subcasting Protocol for Wireless Environments*; SBRC - Simpósio Brasileiro de Redes de Computadores (2000);
- [Suh, Shin and Kwon 2001] Suh Young-Joo, Shin Hee-Sook, and Kwon Dong-Hee, "An Efficient Multicast Routing Protocol in Wireless Mobile Networks" *ACM Wireless Networks*, vol.7, no.5, pp.443-453, Sep. 2001.
- [Xylonmenos and Polyzos 1997] Xylonmenos G. and Polyzos G., IP multicast for mobile hosts, *IEEE Communications Magazine* (January 1997) 54–58.

SOBRE OS AUTORES

Renato Maia é Mestre em Informática pela PUC-Rio e Bacharel em Ciência da Computação pela UFPA. Atualmente é analista do laboratório de tecnologia em computação gráfica da PUC-Rio (TeCGraf). Suas áreas de pesquisa incluem Sistemas Distribuídos, Adaptação Dinâmica e Linguagens de Programação.

Vagner Sacramento é doutorando em informática pela PUC-Rio, Mestre em Sistemas e Computação pela UFRN, Bacharel em Sistemas de Informação pela Ulbra. Professor da disciplina Redes de Computadores no curso de especialização APGS na CCE/PUC-Rio. Experiência como consultor na área de redes e segurança. Áreas de pesquisa de interesse: Protocolos e Sistemas Distribuídos, Computação Móvel, Middlewares, Colaboração entre usuários Móveis, Segurança de Redes de Computadores.

Markus Endler é Bacharel em Matemática e Mestre em Informática pela PUC-Rio (1985 e 1987), e obteve os títulos de Dr.rer.nat. pela Universidade Técnica de Berlim (TU Berlin) em 1992, e de Professor livre-docente da Universidade de São Paulo em 2001. De 1994 to 2000 ele foi professor do Instituto de Matemática e Estatística da USP, e desde 2001 ele é professor em tempo integral do Departamento de Informática da PUC-Rio, onde ele coordena o Laboratory for Advanced Collaboration (LAC). Suas principais áreas de pesquisa são Sistemas e Protocolos Distribuídos, Computação Móvel e Ubíqua, Middlewares, Serviços baseados em localização e Colaboração Móvel. É membro da SBC e da ACM, e participa de comitês técnicos de vários simpósios e workshops nacionais e internacionais em Redes e Sistemas Distribuídos.

Chamada Remota de Métodos na Plataforma J2ME/CLDC

Fernando M. Q. Pereira¹ Roberto S. Bigonha¹
Marco Túlio O. Valente² Mariza A. S. Bigonha¹

¹Dept. de Ciência da Computação - Universidade Federal de Minas Gerais

²Dept. de Ciência da Computação - Pontifícia Universidade Católica de Minas Gerais

Resumo - Este artigo apresenta o sistema RME, uma plataforma de middleware que fornece aos desenvolvedores de aplicações distribuídas um serviço de invocação remota de métodos. RME foi desenvolvido sobre o perfil MIDP/CLDC da plataforma J2ME, a versão da linguagem Java voltada para dispositivos limitados computacionalmente, como aparelhos celulares e palmtops. A implementação de RME, baseada em uma série de conhecidos padrões de projeto, oferece aos desenvolvedores diversas opções de reconfiguração, o que a torna adequada tanto para pequenos dispositivos quanto para computadores dotados de recursos mais poderosos.

Abstract - This paper presents the RME system, a middleware platform that aims to provide application developers with a remote method invocation service. The platform has been developed to the J2ME version of the Java language, which is directed to small devices like cell phones and palmtops. More specifically, it targets a configuration of J2ME called CLDC. Its implementation uses several design patterns which make it easy to reconfigure the system in order to address different requirements imposed by the execution environment. Because of this approach, RME can be used in limited resource devices as well as in more powerful computers.

I. INTRODUÇÃO

Aplicações voltadas para computadores móveis como PDA's e telefones celulares são cada vez mais demandadas. Recentes avanços na tecnologia empregada em tais dispositivos contribuíram para diminuir o custo dos mesmos e torná-los acessíveis a um grupo maior de pessoas. A título de exemplo, estima-se que existam mais de um bilhão de assinantes de serviços de comunicação sem fio em todo o mundo [13]. Esses avanços também contribuíram para aumentar a capacidade computacional de dispositivos móveis, de modo que, atualmente são comuns aparelhos celulares capazes de utilizar informações disponíveis na Internet ou que podem comportar-se como clientes em aplicações de comércio eletrônico.

Ao contrário do mercado de computadores pessoais, para os quais existem poucos fabricantes, o universo dos dispositivos móveis, como telefones celulares, é extremamente segmentado. Existem diversos fabricantes e também vários padrões de comunicação, por exemplo, GSM, TDMA e CDMA. A grande variedade de processadores e protocolos existentes no mundo da computação sem fio compromete a portabilidade de

programas entre dispositivos móveis e constitui um obstáculo aos desenvolvedores de aplicações, pois, sem um padrão universalmente adotado, dificilmente um aplicativo projetado para determinado tipo de processador poderá ser utilizado em um aparelho diferente sem necessitar de modificações em seu código executável. A fim de amenizar os problemas causados pela grande variedade de fabricantes e protocolos existentes e prover aos projetistas de aplicações um modelo de desenvolvimento comum, foi desenvolvida a plataforma J2ME[13], um ambiente de execução Java que requer menos de um décimo dos recursos necessários ao sistema Java tradicional, conhecido como J2SE [1].

Dentre as principais características da plataforma J2ME, que a tornam adequada a uma ampla gama de dispositivos móveis, destacam-se seu tamanho reduzido, a portabilidade e a facilidade de escrita e manutenção de código que proporciona. O tamanho reduzido da plataforma é uma característica importante porque, mesmo com os recentes avanços na tecnologia de computação móvel, dispositivos sem fio são ainda limitados computacionalmente, sendo que muitas vezes não dispõem de mais que algumas dezenas de *kilobytes* de memória. Em segundo lugar, dada a grande variedade de dispositivos móveis existentes, portabilidade é essencial, pois permite a reutilização de código entre diferentes processadores e facilita a comunicação entre sistemas distribuídos. Por fim, Java, sendo uma linguagem de alto nível, orientada por objetos, possui melhores mecanismos de abstração que outras linguagens e ferramentas tradicionalmente utilizadas para o desenvolvimento de aplicações para sistemas embutidos, como C, por exemplo, o que facilita o projeto e manutenção de sistemas de *software*.

Conforme será descrito na Seção II, a plataforma J2ME apresenta diversas configurações, cada uma delas específica para um determinado grupo de processadores. A configuração destinada a aparelhos de telefonia celular, denominada CLDC[13], é muito simples, e, embora disponibilize aos desenvolvedores de aplicações uma ampla variedade de funcionalidades, não contém muitos dos serviços tradicionalmente encontrados em outras versões da linguagem Java, como o mecanismo de Invocação Remota de Métodos.

Java RMI (*Java Remote Method Invocation*) [21], é um modelo de objetos distribuídos desenvolvido para a linguagem Java que pode ser visto como uma evolução do serviço de RPC (*Remote Procedure Call*) [20]. Java RMI estende o modelo de objetos da linguagem Java, permitindo que objetos localizados em espaços de endereçamento de diferentes máquinas virtuais possam interagir entre si. Ao permitir que componentes

distribuídos possam ser utilizados como se estivessem todos localizados no mesmo espaço de endereçamento, Java RMI disponibiliza aos desenvolvedores de aplicações uma poderosa abstração, pois eles podem se ater aos detalhes do modelo de objetos a ser desenvolvido sem, contudo, preocupar-se com detalhes de comunicação inerentes a uma rede de computadores.

Embora o modelo de invocação remota de métodos adotado nas versões J2SE e J2EE da linguagem Java disponibilize aos seus usuários algumas opções de reconfiguração, sua arquitetura ainda é bastante monolítica. Isto quer dizer que muitos dos componentes do sistema Java RMI não podem ser removidos ou alterados, o que torna o modelo inadequado para plataformas dotadas de poucos recursos, como a configuração CLDC da plataforma J2ME.

O presente trabalho descreve RME (*RMI for J2ME*), um serviço de chamada remota de métodos desenvolvido para a configuração CLDC. O sistema proposto permite que programadores utilizem a mesma sintaxe normalmente utilizada pelo serviço Java RMI presente em outras versões da linguagem. A Seção II expõe as principais características da plataforma J2ME. Na Seção III, por sua vez, é descrita a arquitetura do sistema RME, sendo abordados pontos como os padrões de projeto utilizados em sua implementação e o protocolo de comunicação adotado. Na Seção IV são enumeradas algumas das plataformas de *middleware* que possuem características semelhantes ao sistema RME e, finalmente, a Seção V contém conclusões e descrições de possíveis linhas de pesquisa que podem ser desenvolvidas a partir do trabalho apresentado.

II. A PLATAFORMA J2ME

A linguagem Java possui três distribuições principais, ou edições. Dessas edições, a mais popular é conhecida como J2SE, ou *Java 2 Standard Edition*. A segunda distribuição da linguagem, denominada J2EE [19], ou *Java 2 Enterprise Edition*, foi desenvolvida para atender aplicações que demandam grande robustez e segurança, sendo muitas vezes executadas em servidores de grande capacidade, como, por exemplo, aqueles voltados para serviços de comércio eletrônico. Finalmente, para o mercado constituído por dispositivos de menor capacidade computacional existe a plataforma J2ME, ou *Java 2 Micro Edition*. Cada uma dessas edições define um conjunto de tecnologias que podem ser utilizadas para o desenvolvimento de aplicações. São partes de uma plataforma Java a especificação de uma Máquina Virtual, um conjunto de classes relacionadas e as ferramentas necessárias à instalação e configuração de aplicações.

A plataforma J2ME veio suprir as necessidades de um mercado cada vez maior de dispositivos computacionais, que vão desde *paggers* e *palmtops*, até aparelhos televisores com acesso à Internet. Assim sendo, um dos principais objetivos da plataforma J2ME é definir soluções que sejam válidas para todas essas tecnologias e padrões. Com o intuito de classificar e padronizar a enorme variedade de dispositivos existentes no mercado, foram definidos para o ambiente J2ME os conceitos de *configurações* e de *perfis*.

Uma configuração define uma plataforma mínima

para uma determinada categoria de dispositivos móveis. Tais categorias contêm aparelhos com características similares quanto à capacidade de processamento e à memória disponível. Em termos mais concretos, uma configuração define uma máquina virtual, um conjunto mínimo de bibliotecas e os recursos da linguagem Java que estão disponíveis para os dispositivos de uma determinada categoria. A plataforma J2ME possui atualmente duas configurações principais. A primeira configuração, denominada CLDC (*Connected, Limited Device Configuration*), destina-se a dispositivos dotados de mecanismos de transmissão de dados e que possuem entre 160KB e 500KB de memória disponível. Exemplos típicos incluem os telefones celulares digitais e os assistentes pessoais (PDAs). A segunda destas configurações, denominada CDC (*Connected Device Configuration*), está voltada para dispositivos dotados de maior capacidade computacional, com no mínimo 2 MB de memória disponível [13].

Um perfil atende às demandas específicas de uma certa família de dispositivos. Enquanto uma configuração visa aparelhos que possuem recursos de *hardware* semelhantes, um perfil é definido para dispositivos que executam tarefas semelhantes. Ao contrário de uma configuração, perfis incluem bibliotecas mais específicas, e vários deles podem ser suportados pelo mesmo dispositivo. Além disso, as classes que fazem parte de um perfil tipicamente estendem aquelas definidas para uma configuração. MIDP, ou *Mobile Information Device Profile*, foi o primeiro perfil definido para a plataforma J2ME, tendo sido lançado em novembro de 1999. Esse perfil foi implementado sobre a configuração CLDC. Também para a configuração CDC foi desenvolvido um perfil, o qual ficou conhecido como *Foundation Profile*[13].

A fim de prover a plataforma J2ME com um ambiente no qual aplicações pudessem ser executadas, foi desenvolvida uma máquina virtual específica, denominada KVM. Esta máquina virtual foi desenvolvida para executar em dispositivos dotados de um processador de 16 ou 32 bits e que não dispõem de mais que algumas centenas de *kilobytes* de memória. Essas especificações aplicam-se a uma vasta gama de aparelhos, como por exemplo telefones celulares digitais e *paggers*, dispositivos de áudio e vídeo portáteis e também pequenos terminais de consulta ou pagamento de débitos.

Tendo sido projetado para dispositivos limitados em termos de capacidade computacional, o perfil CLDC possui diversas restrições que não estão presentes em outras versões da linguagem Java. Dentre estas limitações cita-se a não existência do tipo ponto flutuante, utilizado para representar números reais. Outra limitação importante é a ausência de suporte aos mecanismos de reflexividade presentes na linguagem Java, os quais permitem que informações sobre a estrutura interna de um programa, por exemplo, o tipo dinâmico de um objeto, sejam conhecidas enquanto o mesmo está sendo executado. Uma consequência desta última limitação é a impossibilidade de utilizar nesse ambiente o pacote Java RMI, pois este faz uso de reflexividade para passar objetos serializados como parâmetros de invocações remotas ou como valores de retorno das mesmas [9].

III. ARQUITETURA DO SISTEMA DESENVOLVIDO

A arquitetura de RME é baseada em uma série de padrões de projeto, os quais são descrições de soluções normalmente adotadas para alguns problemas comuns em programação [6]. Dentre os padrões utilizados para o desenvolvimento de RME podem-se citar fábricas de objetos, decoradores e *proxies*. Além destes, padrões específicos para sistemas de objetos distribuídos também foram usados, como por exemplo, *reactor* e *acceptor-connector*. O uso de tais padrões será descrito com maiores detalhes nesta e nas próximas seções.

O sistema RME foi desenvolvido com o intuito de poder ser facilmente configurado por desenvolvedores de aplicações de acordo com suas necessidades. A fim de alcançar tais objetivos, muitos dos objetos que compõem a plataforma não são instanciados diretamente, mas a partir de fábricas de objetos, as quais constituem um típico padrão de projeto [6]. Uma fábrica cria uma dentre várias possíveis instâncias de uma classe de acordo com os dados providos para ela. Esse padrão facilita a reconfiguração de RME pois, para alterar um componente do sistema, basta que o desenvolvedor modifique a fábrica responsável por sua criação.

III.1. COMUNICAÇÃO ENTRE APLICAÇÕES E OBJETOS REMOTOS

Em RME, um objeto cliente utiliza representações de objetos remotos para realizar chamadas de métodos sobre eles. Tais representações locais, ou *proxies* [6], são denominadas *stubs* [9] em Java RMI. Este mesmo nome é utilizado em RME. Um *stub* possui a mesma interface que o objeto remoto que ele representa. Cada um de seus métodos se encarrega de converter os argumentos passados em uma seqüência de *bytes* e enviá-la para uma entidade capaz de executar tais métodos com os parâmetros dados. Além de enviar os parâmetros de uma invocação, o *stub* se encarrega de receber o valor que resulta da execução da mesma.

Um *stub* não se comunica diretamente com o objeto remoto que representa. Na realidade, tanto em Java RMI quanto no sistema RME, o código de objetos que serão utilizados remotamente não precisa ser diferente do código de objetos que somente serão utilizados localmente. Assim, para permitir a comunicação entre um objeto remoto e o correspondente *stub*, no modelo RME existem três intermediários entre estas duas entidades, ambos localizados no espaço de endereçamento do objeto remoto. O primeiro deles é o servidor responsável por receber as mensagens enviadas pelos *stubs* e inserir tais requisições em uma fila de prioridade. O segundo intermediário é um objeto escalonador, cuja função é retirar as requisições da fila, de acordo com a política de prioridade adotada e repassá-las para o terceiro e último intermediário. Este último trata-se do elemento responsável por permitir a comunicação entre o objeto remoto e o escalonador anteriormente descrito. Tal objeto, tanto em Java RMI quanto em RME, é denominado *skeleton* [9] e sua estrutura pode ser descrita por um padrão de projeto conhecido como adaptador [6]. A Figura 1 descreve as principais entidades envolvidas no

processo de comunicação entre uma aplicação e um objeto remoto.

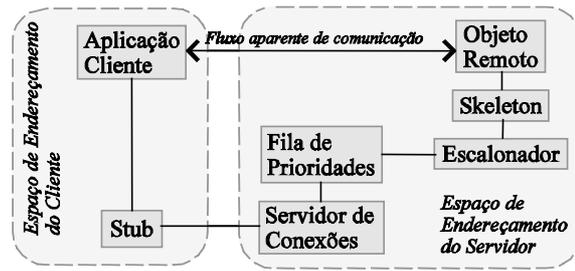


Figura 1. Principais entidades envolvidas na comunicação entre uma aplicação e um objeto remoto

O protocolo utilizado em RME denomina-se RMEP e está descrito na Seção III.7. Para o transporte de seqüências de *bytes* que constituem mensagens desse protocolo foi definida a classe *TcpSocketChannel*. Essa classe, que representa um canal de comunicação, contém métodos para o estabelecimento de conexões e para o envio e recebimento de *bytes*. Para o recebimento de conexões foi definida a classe *TcpSocketServer*.

Caso seja necessário acrescentar funcionalidades adicionais ao canal, como compactação de mensagens, por exemplo, podem ser usadas classes decoradoras. Decoradores [6] constituem um padrão de projeto usado para modificar o comportamento de objetos sem a necessidade de estendê-los via herança. Um decorador pode interceptar chamadas de métodos sobre o objeto decorado, realizar algum processamento sobre essas chamadas baseado em seus parâmetros e repassá-las ao destino original. Para decorar canais de comunicação, RME define a classe *ChannelDecorator*. Esta é a classe ancestral de todos os decoradores de canais, além de ser, ela própria, uma descendente e uma classe cliente de *TcpSocketChannel*. Diversos decoradores, descendentes de *ChannelDecorator*, podem ser agrupados para acrescentar capacidades independentes ao mesmo canal.

A Figura 2 mostra o relacionamento entre o canal de comunicação e o decorador utilizado em RME. A implementação de *TcpSocketChannel* utiliza um *buffer* para tornar mais eficiente o aproveitamento da rede de comunicação, o que permite que somente mensagens completas sejam enviadas. Essa capacidade melhora o aproveitamento do canal, pois pacotes contendo poucos *bytes* deixam de existir. O segundo decorador implementa RMEP, o protocolo descrito na Seção III.7. Caso seja necessário adicionar outras funcionalidades ao canal de comunicação, o usuário de RME pode definir decoradores adicionais para o mesmo.

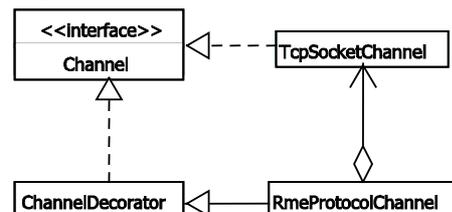


Figura 2: Canal de comunicação e decoradores em RME

Tanto decoradores quanto a extensão via herança permitem agregar novas funcionalidades a componentes

já implementados, ou alterar o comportamento dos mesmos. Contudo, decoradores são um mecanismo mais flexível [11], uma vez que vários deles podem ser combinados de diversas formas diferentes, o que não se verifica em hierarquias de classes. Por exemplo, o arranjo de classes mostrado na Figura 2 poderia ser alterado adicionando-se ou removendo-se um decorador, sem haver a necessidade de modificar o código dos demais componentes. Por outro lado, em uma hierarquia de classes na linguagem Java um novo componente somente pode ser inserido via a extensão da classe mais específica.

III.2. ESTABELECIMENTO DE CONEXÕES

O tratamento de conexões, em RME, utiliza um padrão de projeto denominado *acceptor-connector* [17]. A principal vantagem deste padrão é separar de forma bem definida o estabelecimento de uma conexão do uso da mesma. Uma segunda vantagem é tornar mais simples a realização de chamadas assíncronas, pois esse padrão de projeto permite que a aplicação não seja bloqueada durante o estabelecimento de uma conexão. Tal estabelecimento se dá entre duas entidades, o *conector* e o *receptor*. Conectores representam a parte ativa no processo, uma vez que são responsáveis pelo seu início. Receptores, por sua vez, são entidades passivas, pois executam atividades mediante a chegada de requisições de conexões.

Quando uma requisição de conexão é verificada pelo receptor, um canal é criado e entregue a um objeto responsável por processar as mensagens que serão enviadas por ele. Este componente, denominado *service handler*, ou processador de serviço, é responsável pela troca de mensagens e pelo fechamento do canal, quando o mesmo não for mais necessário. É interessante ressaltar que as duas partes que se comunicam não precisam necessariamente utilizar os mesmos processadores de serviço. Em RME são definidos quatro desses processadores, cuja relação com as demais classes envolvidas neste padrão de projeto é mostrada na Figura 3.

Para enviar requisições de chamadas remotas para servidores, *stubs* utilizam um processador de serviços denominado *RequestSender*. Servidores, por sua vez, utilizam um processador de serviços denominado *RequestReceiver* para receber chamadas remotas e outro, instância da classe *ResponseSender*, para enviar a resposta. Finalmente, o *stub* utiliza uma instância da classe *ResponseReceiver* para receber o valor de retorno ou confirmação de uma invocação.

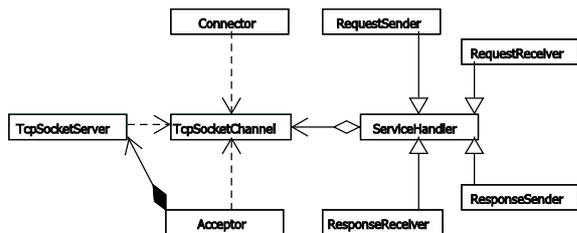


Figura 3. Principais classes envolvidas no estabelecimento de conexões em RME

A relação entre esses quatro elementos é mostrada na

Figura 4. As estruturas apresentadas nessa figura serão discutidas na Seção III.3.

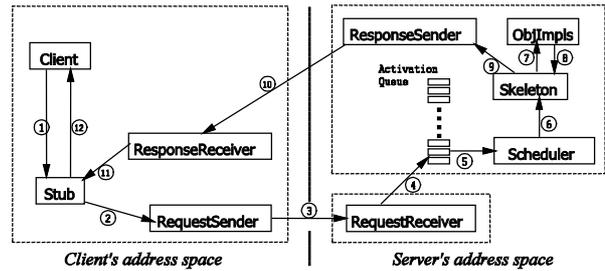


Figura 4. Entidades envolvidas no processo de comunicação entre stubs e skeletons em RME. A região pontilhada indica a área de execução de uma thread e os números indicam a ordem em que métodos são executados durante a invocação remota

A utilização de quatro processadores de serviço facilita o processo de reconfiguração de RME. Por exemplo, caso necessário remover a fila de requisições apresentada na Figura 4, basta alterar a implementação das classes *RequestReceiver* e *RequestSender*. Uma vez que todos os processadores de serviço são criados via fábricas de objetos, tal modificação é simples, não tendo impacto sobre outros componentes do sistema.

III.3. A ESTRUTURA DO SERVIDOR

Chamadas remotas são tratadas por um servidor de requisições cuja estrutura foi parcialmente descrita na Seção III.1. A estrutura de tais servidores segue um padrão de projeto conhecido como *active object* [18]. Este padrão separa o processamento necessário à invocação de um método do processamento necessário à sua execução a fim de simplificar o acesso a um objeto que reside em sua própria *thread* de controle. Na Figura 5 estão representados os principais componentes deste padrão.

De acordo com o padrão *active object*, invocações de métodos não são feitas diretamente sobre o objeto responsável por executá-las. Ao invés disso, tais chamadas são inseridas em uma fila de prioridade. A fim de representar uma chamada remota, foi definida uma classe denominada *MethodRequest*. A fila de prioridade recebe instâncias desta classe, que contêm informações tais como os argumentos da chamada, um identificador que especifica o objeto remoto ao qual tal chamada se refere e a prioridade da requisição. Em RME, a fila de prioridade utiliza um *heap* [4] para efetuar as inserções e remoções de elementos, a fim de permitir que tais operações sejam realizadas de forma eficiente.

Assim como a plataforma CORBA [7], o sistema RME utiliza uma semântica conhecida como *at most once* para caracterizar o processamento de chamadas remotas. Isto quer dizer que, dada uma invocação remota de método, esta é executada no máximo uma vez, podendo, eventualmente, não ser processada. A fim de garantir esta semântica, para cada requisição remota é criado um identificador único, o qual é armazenado na fila de prioridade. Requisições que apresentem identificadores já encontrados são descartadas.

Conforme descrito na Seção III.1, as requisições

armazenadas na fila de prioridade são processadas por uma entidade conhecida como escalonador. Cada servidor remoto, em RME, possui somente um escalonador, embora este possa utilizar diversas *threads* para o processamento de chamadas. Políticas comuns são a criação de uma *thread* para cada requisição que chega à fila de prioridade ou o uso de uma única *thread* para processar todas as chamadas remotas, sendo esta última alternativa a correntemente utilizada em RME, embora ambas tenham sido implementadas.

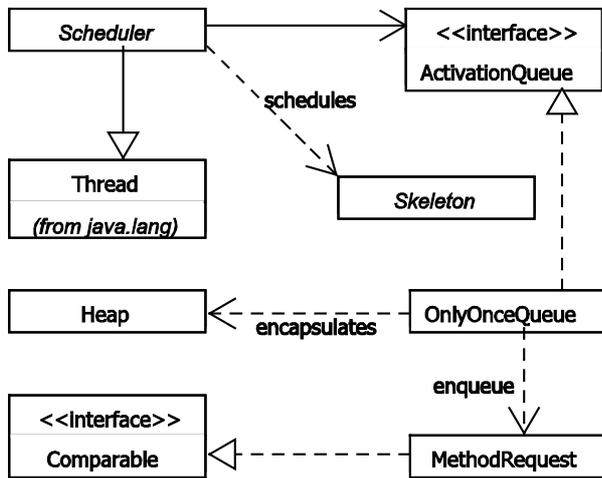


Figura 5. Principais classes que compõem um servidor de chamadas remotas

Um escalonador possui uma coleção de referências para *skeletons*, os quais se encarregam de passar para os objetos remotos as requisições recebidas. *Skeletons* podem ser conectados ao escalonador dinamicamente. Entretanto, caso uma requisição seja retirada da fila em um momento em que o escalonador não possua referência para um *skeleton* capaz de tratá-la, a mesma é descartada. Nesse caso, o escalonador se encarrega de transmitir uma exceção para o endereço de retorno da requisição descartada. A fim de garantir que cada requisição seja tratada pelo objeto ao qual se destina, elas são acrescentadas com o identificador de tal objeto, o qual é único em uma aplicação distribuída.

III.4. CACHES PERSISTENTES E FLYWEIGHT STUBS

Para diminuir o número de conexões e melhorar o aproveitamento da rede sem fio, *stubs* podem ser acrescentados com um *cache* que associa os argumentos de uma chamada remota ao valor retornado pela mesma. Com este fim, o usuário de RME pode definir um decorador de *stubs* que intercepta as invocações, armazenando seus parâmetros e o correspondente valor produzido em uma tabela. Antes de enviar qualquer solicitação remota, esta tabela é verificada e, caso a chamada em questão já tenha sido processada, o resultado da execução anterior é retornado sem que qualquer troca de mensagens aconteça. Ao definir decoradores deste tipo, o usuário deve somente interceptar chamadas que não originem efeitos colaterais, isto é, que não causem modificações no estado do objeto remoto e que sempre retornem valores iguais para os mesmos argumentos passados.

Restrições quanto ao consumo de energia podem levar à interrupção de uma aplicação baseada na plataforma RME. Neste caso, é possível que os valores armazenados em *caches* de *stubs* sejam salvos em memória persistente. Devido a este fato, as fábricas de *stubs*, antes de criar tais objetos, sempre procuram pela existência de arquivos contendo dados a fim de compor *caches*. Caso tais arquivos existam, as novas instâncias de *stubs* são criadas com os dados encontrados.

A pequena quantidade de memória disponível nos dispositivos para os quais se destina a plataforma RME pode comprometer a utilização dos *caches* anteriormente descritos. A fim de contornar este problema, RME utiliza o padrão de projeto conhecido como *flyweight* [6] a fim de reduzir o número de instâncias diferentes de *stubs* em uma aplicação distribuída. Este padrão procura fatorar as informações comuns a diversos objetos em um único componente. Os demais dados, que tornam tais objetos únicos, passam a ser utilizados como parâmetros de métodos. Em RME este padrão é utilizado de modo a assegurar que em uma aplicação exista somente uma instância de *stub* para um dado objeto remoto. Isto quer dizer que, se em uma aplicação distribuída existem diversas referências remotas para o mesmo objeto, todas elas compartilham um *stub* comum.

Tanto a utilização de *caches* em *stubs* e a possibilidade de armazená-los de forma persistente quanto a utilização do padrão *flyweight* visando a economia de recursos são mecanismos opcionais em RME. Quaisquer destas técnicas podem ser usadas de forma independente ou em conjunto, desde que a fábrica de *stubs* apropriada seja utilizada pela aplicação.

III.5. OBJETOS REMOTOS

Em RME, objetos remotos, isto é, objetos cujos métodos poderão ser invocados remotamente, devem estender a classe *RemoteObject*. Tal classe é responsável por definir a semântica de algumas operações comuns a objetos na linguagem Java, como *equals*, *hashCode* e *toString*. *RemoteObject* implementa uma interface que define dois métodos responsáveis pela forma como um servidor de métodos remotos deve ser inicializado e finalizado. São estes os métodos *activateObject* e *deactivateObject*. Na classe *RemoteObject*, tais métodos foram declarados abstratos a fim de aumentar as possibilidades de reconfiguração de RME. O método *activateObject* é responsável por criar um *skeleton* para o objeto remoto e registrar este *skeleton* em um escalonador. Além disso, *activateObject* inicializa um receptor de conexões para receber as requisições destinadas ao objeto remoto. O método *deactivateObject*, por sua vez, tem a função de remover do escalonador o *skeleton* correspondente ao objeto remoto em questão e interromper o receptor de conexões anteriormente criado.

A semântica de referências remotas é definida em RME pela classe *RemoteRef*. Objetos desse tipo permitem que *stubs*, localizados no espaço de endereçamento de aplicações clientes, possam fazer referência aos objetos remotos que representam. Uma referência remota contém duas informações: um identificador que caracteriza de modo único o objeto remoto que lhe corresponde e um

endereço dado pelo par (*endereço IP, número de porta*). Tais informações são, respectivamente, representadas pelas classes *RmeIdentifier* e *HostPortEpid*.

A Figura 6 mostra as iterações existentes entre as principais classes descritas nesta seção. A interface *Marshalable* caracteriza, em RME, objetos que podem ser transformados em seqüências de *bytes* e enviados via um canal de comunicação. Esta interface será descrita na Seção III.6.

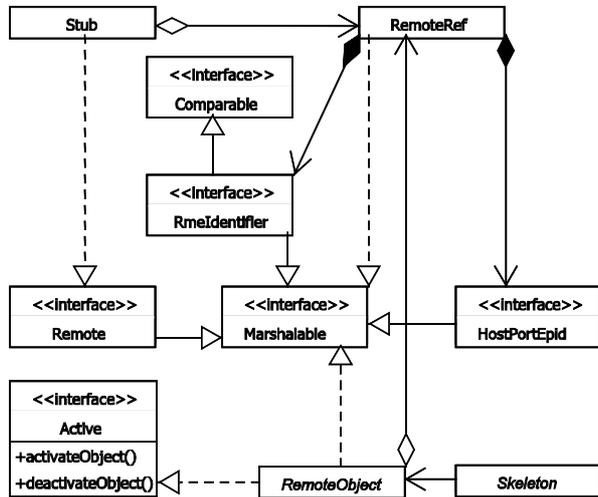


Figura 6. Iteração entre objeto remoto, referência remota, stub e skeleton

III.6. PASSAGEM DE OBJETOS POR VALOR

Conforme descrito na Seção I, em J2ME objetos não podem ser serializados segundo os mesmos princípios utilizados nas outras versões da linguagem Java, devido à ausência, nessa plataforma, dos mecanismos de reflexividade encontrados naquelas edições. A fim de contornar esse problema e permitir que objetos sejam passados como parâmetros ou como valores de retorno de chamadas remotas, sem, contudo, comprometer características da linguagem como polimorfismo e sobrecarga, em RME foi definida uma interface denominada *Marshalable*. Esta interface contém dois métodos: *marshal* e *unmarshal*, que definem, respectivamente, como um objeto pode ser transformado em uma seqüência de *bytes* e como pode ser recuperado a partir de uma seqüência desse tipo. Em RME, apenas objetos que implementam essa interface podem ser enviados por um canal de comunicação.

A solução adotada em RME transfere para o desenvolvedor de aplicações a tarefa de serializar os objetos, porém define um protocolo de serialização. Com este fim, em RME foi implementada a classe *RmeStream*, a qual contém rotinas para converter em *bytes* tipos primitivos como inteiros e valores booleanos, além de tipos compostos como arranjos de objetos e *Strings*. Para a serialização de objetos, *RmeStream* implementa os métodos *readObject()* e *write(Marshalable)*, os quais fazem uso dos métodos *marshal* e *unmarshal* implementados pelo usuário.

O método *write* utiliza o método *marshal* do objeto a ser serializado para gerar uma seqüência de *bytes* que o representa e anota tal cadeia com o nome da classe a que

pertence o objeto. O processo inverso, de criação de um objeto a partir da seqüência de *bytes* que o descreve, faz uso desse nome para obter uma instância da classe nomeada. Para preencher os atributos da instância assim obtida, o método *unmarshal* da mesma é então aplicado à cadeia formada pelos *bytes* remanescentes.

Objetos que descrevem exceções também podem ser serializados segundo o protocolo de serialização definido em RME desde que implementem a interface *Marshalable*. Isto é necessário porque caso um método remoto ative uma exceção enquanto estiver sendo processado, o objeto cliente precisa ser notificado da ocorrência do erro. A fim de permitir esta notificação, exceções levantadas por um método remoto sempre são passadas de volta para o cliente no lugar do valor de retorno esperado. A extração de um objeto que denota uma exceção força o seu disparo, o que permite que erros gerados em espaços de endereçamento remotos interfiram no fluxo de execução de aplicações clientes.

III.7. PROTOCOLO DE COMUNICAÇÃO

O protocolo JRMP [9], utilizado pela plataforma Java RMI para o transporte de dados, é pouco adequado ao ambiente no qual executam computadores móveis [3]. São duas, as causas principais de tal inadequação. Inicialmente cita-se o fato de JRMP utilizar mensagens de tamanho considerável (em termos de número de *bytes*). A segunda destas causas é o fato do protocolo utilizar diversas mensagens, entre requisições e confirmações, para permitir que objetos distribuídos se comuniquem. Por exemplo, a localização de um objeto distribuído e a realização de uma chamada remota sobre um de seus métodos, em RMI, exigiria entre seis e oito trocas de mensagens, ao passo que destas, somente duas mensagens conteriam informação pertinente à invocação remota propriamente dita [3]. Dispositivos móveis utilizam conexões de capacidade limitada e normalmente intermitentes, e, por isso, requerem um protocolo de comunicação mais otimizado.

A plataforma RME utiliza um protocolo de comunicação denominado RMEP, ou *RME Protocol*, o qual foi desenvolvido com o objetivo de utilizar o menor número possível de mensagens para permitir a comunicação entre objetos distribuídos. A fim de reduzir o número de mensagens necessárias a uma chamada remota, RMEP não utiliza mensagens do tipo *Ping* para verificar se conexões existentes ainda podem ser utilizadas, como é feito em Java RMI [3]. Em RME novas conexões são criadas sempre que uma invocação remota precise ser executada e são destruídas ao término da mesma. Este padrão de criação de conexões é dispendioso, pois invocações remotas do mesmo método sobre o mesmo servidor originam o estabelecimento de novos canais, sem que recursos já criados sejam reaproveitados. Embora tal abordagem possa parecer inadequada, é preciso lembrar que aplicações em computadores móveis tendem a ser executadas durante curtos períodos de tempo, e normalmente apenas para pequenas consultas, de modo que poucas conexões tendem a existir simultaneamente em máquinas clientes.

O protocolo RMEP utiliza quatro tipos de mensagens diferentes, as quais foram nomeadas *call*, *return*, *ping* e *ack*. Mensagens do tipo *call* caracterizam as invocações remotas de métodos e são respondidas por mensagens do tipo *return*, as quais contêm valores que resultam da execução de algum método remoto. As mensagens do tipo *ping* e *ack* são utilizadas para verificar se existem servidores de métodos remotos ativos em um dado endereço.

A Tabela 1 contém uma gramática que define o formato das mensagens utilizadas por RMEP. Nesta gramática, os símbolos *call*, *ping*, *return* e *ack* denotam constantes que caracterizam o tipo da mensagem, as quais foram separadas em duas categorias. A primeira delas é composta por mensagens do tipo *call* e *ping*. Tais mensagens exigem uma resposta, de acordo com o protocolo RMEP, e, por isso, incluem um endereço para retorno. O símbolo *epid* representa a seqüência de *bytes* obtida da serialização de um objeto do tipo *HostPortEpid*, o qual contém o nome do *host* e o número da porta que definem o endereço para o qual a mensagem de resposta deve ser enviada. Identificadores de chamadas remotas e identificadores de objetos são representados na gramática dada, respectivamente, pelos símbolos *call_{id}* e *obj_{id}*. Para a manipulação de tais identificadores, a plataforma RME define a classe *RmeIdentifier*, representada, na Tabela 1 pelo símbolo *id*. Finalmente, os argumentos de uma chamada remota e o valor de retorno da mesma são representados pelo símbolo *byteStream*, que denota uma seqüência de *bytes* obtida da serialização de objetos e de tipos primitivos.

```

Message → header version msgtype
header  → 0x52 0x4d 0x45 0x50
version → 0x01
msgtype → msgreq | msgack
msgreq  → epid reqtype
reqtype → call called objid op args
          | ping
callid  → id
objid   → id
op       → int
args     → byteStream
msgack  → return callid retval
          | ack
retval  → byteStream

```

Tabela 1: Formato de mensagens do protocolo RMEP

III.8. O SERVIÇO DE LOCALIZAÇÃO DE NOMES

A fim de que o serviço de invocação remota de métodos possa ser plenamente utilizado, é necessário disponibilizar à aplicação uma maneira de localizar objetos distribuídos. Em Java RMI, tal serviço de localização está disponível por meio de uma classe denominada *Naming* [9]. Na plataforma RME foi definida a classe *RmeNaming*, que disponibiliza ao usuário as mesmas funcionalidades encontradas em Java RMI, tais como a associação de nomes a objetos remotos e a busca de objetos remotos a partir dos nomes a que foram associados.

Assim como em Java RMI, também em RME o serviço de localização é controlado por um objeto remoto, o *servidor de nomes*. Em RME tal objeto remoto recebe conexões na porta 1101. A classe *RmeNaming* encapsula

o acesso ao servidor de nomes, disponibilizando aos desenvolvedores de aplicações uma sintaxe mais simples do que a que seria necessária caso referências remotas fossem utilizadas diretamente.

O servidor de nomes utiliza uma *thread* separada para verificar quais referências nele armazenadas permanecem válidas. Esta *thread* envia periodicamente mensagens do tipo *ping*, descritas na Seção III.7, para os objetos que cadastraram referências remotas no servidor de nomes. Os objetos que não responderem às mensagens enviadas em um certo intervalo de tempo têm suas referências removidas da tabela de nomes. Isto evita que referências para componentes que não mais existam sejam enviadas para objetos clientes.

III.9. A API DE RME E UM EXEMPLO DE USO

Aplicações distribuídas que utilizam o pacote RME são sintaticamente semelhantes àquelas desenvolvidas a partir de Java RMI. Assim como em Java RMI, uma interface é utilizada para definir um conjunto de métodos que poderão ser invocados remotamente. Em ambas as plataformas, estas interfaces devem estender uma outra interface, denominada *Remote*, que as caracteriza como passíveis de serem utilizadas remotamente. A Figura 7 exemplifica, em RME, uma interface onde é declarado um método remoto. Esta interface caracteriza um servidor que responde a consultas de números telefônicos. Tal servidor recebe pesquisas contendo nomes de pessoas e retorna uma cadeia de caracteres representando os números telefônicos associados aos nomes recebidos.

Observe que o pacote importado pelo programa mostrado na Figura 7 é denominado *arcademis*, e não *rme*. Tal fato se justifica porque a plataforma RME foi implementada como uma instância de um arcabouço para desenvolvimento de *middlewares* orientados por objetos denominado *arcademis*. Maiores detalhes sobre esse arcabouço podem ser obtidos em [12].

```

import arcademis.*;
public interface PhoneBook extends Remote {
    public String getPhone(String name)
        throws ArcademisException;
}

```

Figura 7. Definição de uma interface remota em RME

Objetos remotos, ou seja, as entidades que devem implementar os métodos de interfaces remotas, como a mostrada na Figura 7, precisam estender a classe *RemoteObject*. A implementação destes objetos, contudo, não precisa levar em consideração o fato de seus métodos poderem ser utilizados remotamente. A Figura 8 mostra uma implementação que poderia ser provida para a interface apresentada na Figura 7. No exemplo mostrado, *PhoneAddress* é uma classe utilizada para associar ao nome de uma pessoa suas informações particulares como número telefônico e endereço domiciliar. É importante ressaltar que todas as inicializações relacionadas à semântica distribuída do objeto ficam a cargo do método *activateObject()*, que pertence à classe *RemoteObject* e também o método construtor desta mesma classe. O método construtor, por exemplo, se encarrega de criar um *skeleton* e uma referência remota para o objeto.

Para que um objeto remoto como o mostrado na Figura 8 possa ser utilizado por aplicações distribuídas, é necessário que ele seja registrado em um servidor de nomes e ativado. Um exemplo de aplicação em que tais procedimentos são executados é mostrado na Figura 9.

```
import rme.*;
import arcademis.*;
public class PhoneBook_Impl extends RemoteObject
implements PhoneBook {
    Hashtable h = null;
    public String getPhone(String name) {
        PhoneAddress addr = (PhoneAddress)h.get(name);
        if(addr == null) return null;
        else return addr.getPhoneNumber();
    }
    public void putAddress(PhoneAddress addr) {
        h.put(addr.getName(), addr);
    }
}
```

Figura 8. Implementação de uma interface remota

```
import rme.*;
import arcademis.*;
public class PhoneBookServer {
    public static void main(String[] args) {
        RmeConfigurator c = new RmeConfigurator();
        c.configure();
        try {
            PhoneBook_Impl phoneList = new
            PhoneBook_Impl();
            RmeNaming.bind("phone_book", phoneList);
            phoneList.activateObject();
            // put the code to fill the list with phone
            // information here
        } catch (Exception e) {}
    }
}
```

Figura 9. Inicialização de um objeto remoto

```
import rme.*;
import arcademis.*;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class SimpleClient extends MIDlet
implements CommandListener {
    private PhoneBook phoneList = null;
    private Command enterCommand;
    private TextBox tb;
    public SimpleClient() {
        RmeConfigurator mc = new RmeConfigurator();
        mc.configure();
        try {
            phoneList = (PhoneBook)
            RmeNaming.lookup("algol.dcc.ufmg.br\phone_book");
        } catch (Exception e) {}
        enterCommand = new Command("Enter", Command.OK, 1);
        tb = new TextBox("Example", "Send", 15, TextField.ANY);
        tb.addCommand(enterCommand);
        tb.setCommandListener(this);
    }
    protected void startApp() {
        Display.getDisplay(this).setCurrent(tb);
    }
    protected void pauseApp() {}
    protected void destroyApp(boolean u) {}
    public void commandAction(Command c, Displayable d) {
        if (c == enterCommand) {
            try {
                tb.setString(phoneList.getPhone("Lima Barreto"));
            } catch (ArcademisException e) {}
        }
    }
}
```

Figura 10. Aplicação cliente implementada em J2ME

O método `activateObject()` inicializa um servidor responsável por receber requisições remotas e associa o *skeleton* criado pelo método construtor de `RemoteObject` a um escalonador, o qual é responsável por definir a ordem em que invocações são repassadas à implementação do objeto remoto, conforme descrito na Seção III.3.

Uma aplicação precisa definir a configuração da plataforma RME que será utilizada a fim de poder realizar chamadas remotas. Estas definições incluem a escolha das fábricas de objetos que irão compor a camada RME da aplicação. Para este fim, o pacote disponibiliza ao desenvolvedor a classe `RmeConfigurator`, a qual contém algumas opções de configuração que garantem o funcionamento do sistema. Caso seja necessário alterar componentes da plataforma RME, uma extensão da classe configuradora pode ser utilizada. Para a localização de objetos distribuídos, por sua vez, a classe `Naming` implementa o método `lookup`, que retorna, a partir de um nome de objeto, um *stub* para o mesmo. Um exemplo de aplicação para a plataforma J2ME que utiliza o sistema RME é mostrado na Figura 10.

III.10. ANÁLISE DE DESEMPENHO

Para medir o desempenho da plataforma RME foram feitos alguns testes utilizando como cliente um emulador de telefone celular cuja máquina virtual K (KVM), executa 100 *bytecodes* por milissegundo. Tanto o servidor de requisições quanto o emulador de celular foram executados em computadores de igual capacidade: *Pentium 4* de 2.0GHz e 512MB de memória disponível conectados por uma rede *ethernet* de 10Mb/s. Foi medido o número de requisições tratadas por segundo para seis métodos que não possuem argumentos e cujos valores de retorno são diferentes. Na Tabela 2 para cada método, é mostrado o valor médio apurado a partir da execução de 10 séries de 50 invocações remotas.

A fim de fornecer um limite superior para o maior número de requisições que podem ser tratadas no cenário anteriormente descrito, foi implementada uma aplicação que troca seqüências de *bytes* de mesmo tamanho que aquelas trocadas no teste realizado com RME, sem, contudo, realizar qualquer processamento sobre tais cadeias. Essa aplicação utiliza a mesma implementação de canal utilizada em RME: uma conexão baseada em sockets TCP/IP que utiliza um *buffer* de dados para permitir o agrupamento de *bytes* antes da transmissão dos mesmos. Além disso, assim como em RME, também nessa aplicação uma nova conexão é criada para simular cada invocação remota. Tanto a aplicação baseada em sockets quanto aquela baseada em RME são constituídas por um cliente implementado em J2ME e por um servidor implementado em J2SE. Os resultados apurados são mostrados na Tabela 2.

Pela análise da Tabela 2, nota-se que quanto mais complexo o valor de retorno de uma chamada remota, maior o tempo necessário para completá-la. Tal atraso deve-se ao algoritmo de serialização utilizado na implementação de RME. O tempo de execução desse algoritmo é relevante dadas as características da máquina cliente: um emulador capaz de executar 100 *bytecodes* por

milissegundo. É possível observar também que o tamanho das mensagens transmitidas não é um fator decisivo para o desempenho da aplicação. Esse fato se deve à utilização de um *buffer* na implementação do canal de transmissão, pois tal estrutura permite que todos os *bytes* que constituem uma mensagem sejam enviados em um mesmo pacote de dados.

Valor de retorno	tamanho da resposta (<i>bytes</i>)	req./s RME	req./s sockets
byte	36	4.39	5.00
short	37	4.38	5.00
int	39	4.49	4.98
long	43	4.35	4.99
String	48	3.89	4.99
String[]	168	2.51	4.98

Tabela 2. Testes de desempenho realizados sobre uma aplicação RME

IV. TRABALHOS RELACIONADOS

Existem diversas plataformas *middleware* que utilizam o modelo de programação orientada por objetos no desenvolvimento de aplicações distribuídas. Um dos primeiros exemplos desse tipo de sistema foi implementado na linguagem Modula 3, no início da década de 90 [2]. Vários dos conceitos e algoritmos empregados nessa implementação pioneira, como o mecanismo de coleta de lixo distribuída e as técnicas de serialização de objetos são atualmente utilizados em muitas das plataformas de *middleware* mais populares, como Java RMI, por exemplo.

Plataformas de *middleware* como CORBA [7], .NET [5] ou Java RMI são sistemas complexos e monolíticos e, devido a este fato, sua utilização em ambientes caracterizados pela limitação de recursos fica comprometida. Algumas tentativas de portar tais plataformas para o ambiente móvel foram realizadas, principalmente com relação ao sistema CORBA. Neste contexto, um conjunto mínimo da especificação CORBA foi definido justamente com o intuito de permitir o desenvolvimento de aplicações compatíveis com aquela plataforma e que utilizassem somente os recursos estritamente necessários. Tal subconjunto de CORBA é conhecido como *minimumCORBA* [8].

Existem diversos sistemas de *middleware* que, assim, como RME, permitem reconfiguração estática, como, por exemplo, a plataforma TAO [14], em cuja implementação foram propostos diversos padrões de projeto incorporados na implementação de RME [16]. Este sistema utiliza arquivos contendo descrições de configurações para especificar as estratégias que serão utilizadas, por exemplo, para tratamento de conexões, concorrência e escalonamento, as quais são definidas estaticamente durante a inicialização das aplicações. A principal característica de tais sistemas é o fato de permitirem a implementação de plataformas de *middleware* contendo apenas as funcionalidades necessárias para as aplicações a que se destinam. Por exemplo, é possível configurar uma instância de RME contendo somente o código necessário à parte cliente de uma aplicação, ou somente o código necessário à parte servidora da mesma.

O sistema conhecido como *UIC CORBA* [14], assim

como RME, foi desenvolvido com o objetivo de prover alternativas adequadas ao desenvolvimento de aplicações distribuídas para computação móvel. UIC permite que desenvolvedores de aplicações utilizem opções de reconfiguração estáticas e dinâmicas, isto é, permite que o sistema seja alterado durante a sua execução. Para esse fim, o sistema define um arcabouço de componentes abstratos que encapsulam os aspectos considerados comuns a toda plataforma compatível com CORBA, como os protocolos de transporte e serialização utilizados. Classes concretas, carregadas dinamicamente, implementam essas propriedades de acordo com as necessidades de uma determinada plataforma de *middleware*.

A plataforma *LegORB* [15] é outro sistema de *middleware* compatível com CORBA e que disponibiliza aos desenvolvedores de aplicações grande número de opções de reconfiguração. O sistema permite tanto modelos de configuração estáticos quanto dinâmicos. A versão estática fornece os melhores resultados em termos de utilização de memória. A título de exemplo, plataformas obtidas via configuração estática não ocupam mais que 23KB de memória, ao passo que sistemas dinamicamente configurados ocupam em torno de 140KB de memória.

Os sistemas anteriormente apresentados foram implementados em C++ e são compatíveis com a plataforma CORBA, ao passo que RME é completamente implementado na linguagem Java. Por outro lado, existe uma implementação de RMI para a configuração CDC, apresentada na Seção II, que está disponível como um pacote opcional de J2ME. Este sistema, conhecido como *RMI OP (RMI Optional Package)* [10], permite que aplicações clientes em dispositivos CDC manipulem objetos localizados em servidores remotos. A principal diferença entre RMI OP e RME são as configurações para as quais tais sistemas se destinam. Enquanto RME foi desenvolvido para CLDC, o pacote RMI OP não é suportado por esta configuração.

Os sistemas *LegORB*, *UIC CORBA* e RME têm o objetivo de facilitar o desenvolvimento de aplicações distribuídas para computação móvel. Sendo assim, a utilização eficiente da memória disponível é um requisito essencial que deve ser atendido por essas plataformas. A Tabela 3 mostra o tamanho das bibliotecas que compõem cada um destes sistemas. Essa tabela trata somente das bibliotecas necessárias à parte cliente de uma aplicação distribuída, sendo que, no caso de RME, o espaço ocupado pela máquina virtual Java não está sendo levado em consideração. Caso seja necessário, existem algumas maneiras de reduzir o tamanho do pacote RME. Por exemplo, conforme discutido na Seção 3, a implementação de RME utiliza fábricas para a criação de objetos.

RME	34.5KB
LegORB configurado estaticamente	22.5KB
LegORB configurado dinamicamente	141.5KB
UIC CORBA (PalmOS)	18KB
UIC CORBA (Windows CE(SH3))	29KB
UIC CORBA (Windows 2000)	72KB

Tabela 3. Comparação entre o tamanho das bibliotecas dos sistemas *LegORB*, *UIC CORBA* e RME

Embora tais fábricas facilitem a reconfiguração do sistema, elas suas bibliotecas. A fim de reduzir esse tamanho, as fábricas podem ser removidas e os objetos podem ser instanciados diretamente.

V. CONCLUSÕES E TRABALHOS FUTUROS

Este trabalho apresentou o *middleware* RME, que foi desenvolvido com o intuito de prover um serviço de chamada remota de métodos para o perfil MIDP/CLDC da plataforma J2ME. Essa é a principal contribuição da plataforma RME, uma vez que não existe ainda nenhum sistema desse tipo para CLDC cuja implementação esteja consolidada. Além de descrever os principais detalhes relativos à implementação da plataforma RME, como por exemplo os padrões de projeto utilizados e o protocolo de transmissão de dados, foi descrito um exemplo de aplicação distribuída baseada nos serviços fornecidos pelo sistema e também foram mostradas comparações entre RME e alguns dos sistemas semelhantes mais conhecidos.

A implementação de RME utilizou diversos padrões de projeto, o que deu origem a um sistema bastante reconfigurável. Enquanto os sistemas de *middleware* mais populares possuem um projeto monolítico, RME é constituído por um conjunto de componentes que implementam funcionalidades básicas e que podem ser removidos ou alterados de maneira simples. Dessa forma, o sistema RME pode ser utilizado para definir aplicações clientes, servidoras ou que agregam ambas capacidades, podendo enviar e receber invocações remotas de métodos. Essa propriedade é útil no desenvolvimento de aplicações para telefonia celular porque em muitos casos não há necessidade dessas atuarem como servidores de requisições remotas.

O pacote RME desenvolvido para J2ME não é compatível com a versão J2SE da linguagem Java. Basta, contudo, alterar duas classes para portar o sistema de uma plataforma para a outra. Essas classes, denominadas `TcpSocketChannel` e `TcpSocketServer`, utilizam os mecanismos particulares que cada plataforma Java provê para o estabelecimento de conexões e tais funcionalidades não são as mesmas nos dois ambientes. Dada a utilização de fábricas para a instanciação de objetos essa modificação é trivial. Assim, com o intuito de validar o sistema foram implementados servidores de métodos remotos em J2SE e clientes baseados em J2ME.

As bibliotecas utilizadas por aplicações clientes, as quais incluem o serviço de localização de nomes, os protocolos de transmissão e serialização, as definições de referências remotas e *stubs*, todas as classes necessárias ao estabelecimento e manutenção de conexões, bem como ao envio de dados, além de uma série de funcionalidades para tratamento de erros ocupam cerca de 34.5KB de memória. As bibliotecas utilizadas por aplicações servidoras de chamadas remotas, por outro lado, ocupam cerca de 67.9KB. Tais bibliotecas, além de conterem todas as classes que dão suporte às aplicações clientes, contêm também servidores de conexões, um gerenciador de evento, a fila de prioridade na qual são armazenadas requisições remotas, definições de objetos remotos e *skeletons*, além de um escalonador que determina a ordem em que requisições serão atendidas.

Dentre as características da plataforma RME que a tornam adequadas para a computação móvel, citam-se, em primeiro lugar, o caráter modular do sistema, o qual permite a remoção ou adição de componentes de acordo com as necessidades da aplicação. Além disto, RME utiliza um protocolo otimizado, tanto com relação ao tamanho das mensagens trocadas quanto com relação à quantidade delas, o que se mostra apropriado à pequena banda de transmissão que dispositivos móveis têm disponível. Do caráter reconfigurável da plataforma, advêm outras características que a tornam propensa a ser utilizada no ambiente móvel. Desse modo, conforme descrito na Seção III-4, pode-se utilizar o padrão *flyweight* para limitar o número de instâncias de *stubs* existentes em uma aplicação, quando restrições de memória assim o exigirem. E pode-se, ainda, utilizar *caches* em *stubs* para diminuir o tráfego de mensagens na rede.

Como trabalhos futuros, espera-se prover uma configuração de RME capaz de efetuar chamadas assíncronas de métodos. Esta capacidade é particularmente interessante no contexto da computação móvel, pois, dadas as restrições de energia dos dispositivos utilizados, em muitos casos tais aparelhos não são capazes de se manterem conectados durante todo o tempo de processamento da chamada síncrona. Uma vez que o modelo RME já permite o estabelecimento de conexões assíncronas, acredita-se que a extensão desse mecanismo para permitir o assincronismo de chamadas remotas será simples. Conexões ditas assíncronas não bloqueiam a execução da aplicação caso não possam ser efetivadas de imediato. Nesse caso, novas tentativas de estabelecimento de uma conexão seguem a tentativa inicial até que sua efetivação seja possível.

Além de introduzir chamadas assíncronas ao sistema, espera-se acrescentar algumas possibilidades de reconfiguração dinâmica à plataforma RME. Alguns estudos nesse sentido já foram feitos, como permitir, por exemplo, a adoção de algoritmos de criptografia ou compactação que seriam aplicados às mensagens trocadas sem interromper o funcionamento do sistema. Com esse fim o protocolo de comunicação deveria ser alterado de modo a possibilitar que as partes interessadas pudessem negociar a adoção de tais algoritmos.

Finalmente, pretende-se fornecer para RME a implementação do protocolo JRMP, utilizado por Java RMI. Tal adição permitirá que aplicações desenvolvidas nas duas plataformas possam interagir diretamente. Atualmente, tal iteração somente é possível via classes adaptadoras, as quais se encarregam de permitir a comunicação entre os protocolos RMEP e JRMP. Essa abordagem, todavia, não é prática, pois para cada interface remota é necessário implementar um adaptador.

Código Fonte O pacote RME está disponível para cópia sem restrições no endereço eletrônico:

<http://www.dcc.ufmg.br/llp/arcademis>

Agradecimentos Esta pesquisa foi financiada pela Fapemig e pelo CNPq.

REFERÊNCIAS

- [1] Ken Arnold, James Gosling, and David Holmes. *The {Java} Programming Language*. Addison Wesley, 3rd edition, 2000.
- [2] Andrew Birrell, Greg Nelson, Susan Owicki, and Edward Wobber. Network objects. In *14th Symposium on Operating Systems Principles (SOSP)*, pages 217--230. Software--Practice and Experience, 1993.
- [3] Stefano Campadello, Oskari Koskimies, and Kimmo Raatikainen. Wireless java rmi. In *1th International Enterprise Distributed Object Computing Conference*, pages 114--123. USENIX Association, 2000.
- [4] Thomas Cormen, Charles Leiserson, Ronald Rivest, and Clifford Stein. *Algoritmos -- Teoria e Prática*. Campus, 2nd edition, 2002.
- [5] Microsoft Corporation. Microsoft .NET Development. msdn.microsoft.com/net/ -- última visita: junho de 2003.
- [6] Erich Gamma, Richard Helm, Ralph Johnson and John Vlissides. *Design Patterns Elements of Reusable Object-Oriented Software*. Addison-Wesley, October 1994.
- [7] Frantisek Plasil and Michael Stal. An Architectural View of Distributed Objects and Components in CORBA, Java RMI and COM/DCOM. *Software - Concepts and Tools*, 19(1):14--28, 1998.
- [8] ObjectManagement Group. MinimumCORBA Specification, 002.doc.ece.uci.edu/CORBA/formal/02-08-01.pdf -- última visita: junho de 2003.
- [9] SunMicrosystems Inc. RMI home page. <http://java.sun.com/j2se/1.4.1/docs/guide/rmi/spec/rmiTOC.html> -- última visita: junho de 2003.
- [10] SunMicrosystems Inc. Rmi optional package specification version 1.0. <http://java.sun.com/products/rmiop/> -- última visita: junho de 2003.
- [11] James W. Cooper. *The Design Patterns Java Companion*. Addison-Wesley, 2000.
- [12] Fernando Magno Quintão Pereira. Arcademis's home page. <http://www.dcc.ufmg.br/lp/arcademis> -- última visita: september 2003.
- [13] Roger Riggs, Antero Taivalsaari, and Mark VandenBrink. *Programming Wireless Devices with the Java 2 Platform, Micro Edition*. Addison Wesley, 1th edition, July 2001.
- [14] Manuel Román, Fabio Kon, and Roy Campbell. Reflective Middleware: From Your Desk to Your Hand. *IEEE Distributed Systems Online*, 2(5), July 2001.
- [15] Manuel Román, Dennis Mickunas, Fabio Kon, and Roy Campbell. LegORB and Ubiquitous CORBA. In *IFIP/ACM Middleware'2000 Workshop on Reflective Middleware*. Springer-Verlag, 2000.
- [16] Douglas Schmidt and Chris Cleeland. Applying Patterns to Develop Extensible and Maintainable ORB Middleware. *IEEE communications Magazine -- Special Issue on Design Patterns*, 37(4):54 -- 63, 1999.
- [17] Douglas Schmidt. Acceptor-connector -- an object creational pattern for connecting and initializing communication services. In *European Pattern Language of Programs conference*, July 1996.
- [18] Douglas Schmidt and Greg Lavender. Active object - - an object behavioral pattern for concurrent programming. In *Second Pattern Languages of Programs conference*, September 1995.
- [19] Bill Shannon, Mark Hapner, Vlada Matena, James Davidson, Eduardo Pelegri-Llopert, and Larry Cable. *Java 2 Platform, Enterprise Edition: Platform and Component Specifications*. Addison Wesley, 1th edition, May 2000.
- [20] Richard Stevens. *UNIX Network Programming*, volume 1. Prentice Hall, 2nd edition, 1998.
- [21] A.Wollrath, R.Riggs, and J.Waldo. A distributed object model for the Java system. In *2nd Conference on Object-Oriented Technologies & Systems*, pages 219--232. USENIX Association, 1996.

Fernando Magno Quintão Pereira é mestre em Ciência da Computação pela Universidade Federal de Minas Gerais.

Marco Túlio de Oliveira Valente é professor de Ciência da Computação na Pontifícia Universidade Católica de Minas Gerais.

Roberto da Silva Bigonha é professor titular de Ciência da Computação na Universidade Federal de Minas Gerais.

Mariza Andrade da Silva Bigonha é professora adjunta de Ciência da Computação na Universidade Federal de Minas Gerais.

Integrando Redes *Overlay* e Redes de Sensores Sem fio

Daniel F. Macedo, Leonardo B. Oliveira, Antonio A.F. Loureiro
{damacedo, leob, loureiro}@dcc.ufmg.br

Laboratório de Redes e Sistemas Distribuídos
Departamento de Ciência da Computação
Universidade Federal de Minas Gerais
Av. Presidente Carlos Luz,1275 - Belo Horizonte, MG

Resumo - Redes *Overlay* (RO) são redes construídas sobre redes físicas, com o intuito de migrar parte da complexidade de roteamento para a camada de aplicação. Já Redes de Sensores Sem Fio (RSSF) são redes com sérias restrições computacionais e de energia, altamente susceptíveis a falhas. Identificamos que a partir de um pequeno número de nós mais poderosos, que instanciam nós de uma rede *overlay* e que efetuam monitoração da rede subjacente, podem ser descobertas rotas alternativas que atendem melhor as necessidades das aplicações, em detrimento das rotas padrão. Os resultados demonstram que a estratégia é capaz de melhorar a confiabilidade, mantendo ou mesmo melhorando demais métricas como *jitter* e latência, em um cenário com falhas.

Abstract - *Overlay Networks* are networks established over the underlying network, in order to migrate part of the routing complexity to the application layer. At the same time, *Wireless Sensor Networks* are emergent networks where nodes have severe hardware constraints and are also susceptible to failure. Our approach takes advantage of cluster-heads to instantiate an *overlay network* that monitors the underlying network, discovering alternative routes which meet application demands better than the default routes. Our conclusion is that the scheme not only improves reliability, but also other network metrics such as *jitter* and delay.

Palavras-Chave: roteamento, *overlay*, Redes de Sensores Sem Fio, sensores, confiabilidade, falhas.

I INTRODUÇÃO

Redes *Overlay* (RO) são redes lógicas construídas acima da rede física. Atualmente, ROs são utilizadas para viabilizar aplicações e serviços na Internet. Aplicações destas redes podem ser encontradas em redes *Peer-to-Peer* (P2P) [7, 6], serviços *multicast* [16], Redes de Distribuição de Conteúdo (CDNs) [14] e *Resilient Overlay Networks* (RONs) [2]. Nas ROs, são estabelecidos enlaces lógicos entre nós que pertencem à rede. Estes enlaces podem ou não ser enlaces fim-a-fim na rede física (como mostra a Figura 1).

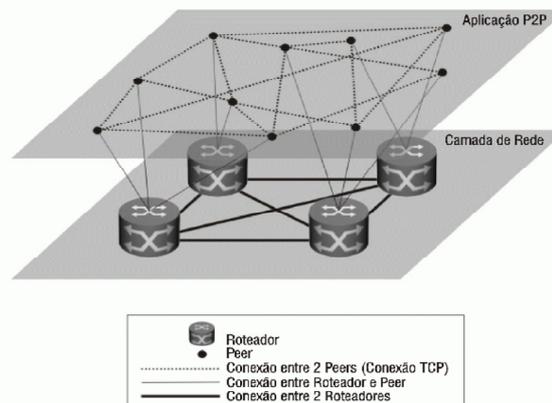


Figura 1: Exemplo de uma rede *overlay*.

As redes infra-estruturadas (como a Internet e as demais redes *IP*), por serem otimizadas para escalabilidade e relativa simplicidade, não são capazes de prover qualidade de serviço com os parâmetros de rede adequados (*jitter*, latência, taxa de entrega, perda, vazão, etc.) para cada aplicação. Uma *Resilient Overlay Network* (RON) é uma rede *overlay* que tem como objetivo prover à aplicação a confiabilidade que falta às redes *IP* atuais [2]. Através da monitoração da rede, as RONs podem identificar, dentre os vários caminhos monitorados para um dado destinatário, qual é o melhor caminho para se enviar dados de uma aplicação específica. Os caminhos monitorados são compostos pelo caminho fornecido pelo algoritmo de roteamento da Internet, bem como outros caminhos, onde um nó da rede *overlay* atua como intermediário no envio de dados.

Redes de Sensores Sem Fio (RSSF) são redes *ad hoc* que possuem, além das características tradicionais deste tipo de rede (auto-organização, inexistência de infraestrutura prévia e pequeno poder computacional), outras peculiaridades [13, 15, 1, 3]. Elas são compostas por um grande número de dispositivos de sensoriamento – isto é, sensores – que são lançados sobre uma região de interesse a fim de colher dados do ambiente e enviá-los para a Estação Base (EB). Os sensores são capazes de exercer o sensoriamento de diferentes características do meio (temperatura, mobilidade, pluviometria e outros), e possuem grandes res-

trições com relação ao custo de fabricação e ao gasto de energia.

Nestas redes, existe um alto grau de falhas, que podem ser causadas por danificação física, término do estoque e energético do sensor, ou mesmo interferência do ambiente. A rede deve se reorganizar a cada falha, para que continue a exercer a sua função, economizando o máximo possível de energia. Devido a estas características, os protocolos de roteamento usuais para redes *ad hoc* são incapazes de atender às necessidades das RSSF [4]. Um sistema de roteamento na camada de aplicação, que é executado somente em nós mais poderosos, permite encontrar rotas não oferecidas pelo protocolo de roteamento. As rotas encontradas levam em consideração a confiabilidade dos enlaces e níveis de energia dos nós que a compõem. O uso destas rotas pode não somente elevar a tolerância a falhas (capacidade de uma rede de sensores de manter suas funcionalidades mesmo quando ocorrem falhas nos nós), como também prolongar a vida útil de toda a rede.

As RSSF podem ser classificadas quanto à composição do roteamento. RSSF planas são redes onde todos os nós podem rotear dados para outros nós da rede. Em redes hierárquicas, os nós sensores são divididos em grupos, onde um dos nós do grupo, chamado de nó *cluster-head*, é responsável por rotear todos os dados produzidos pelos sensores do seu grupo.

O objetivo deste trabalho é apresentar uma RSSF hierárquica que utiliza uma RO para ampliar sua capacidade de tolerância a falhas e aumentar a taxa de entrega da rede.

Os resultados obtidos mostram que a partir desta abordagem é possível não somente melhorar a taxa de entrega de mensagens, bem como também outras métricas de rede, como atraso e *jitter*, mesmo na ocorrência de falhas de nós.

II TRABALHOS RELACIONADOS

Este trabalho foi motivado por diversos projetos de pesquisa, sejam na área de redes *overlay*, sejam na área de confiabilidade em RSSF – entenda-se confiabilidade, aqui, como o grau de confiança de que um pacote enviado alcançará seu destino. Como será visto a seguir, apesar de poucos, já existem trabalhos que utilizam redes *overlay* em ambientes sem fio. Todavia, não se tem conhecimento de algum trabalho que utilizou ROs sobre uma RSSF como um novo modelo de roteamento, seja para fins de melhoria de métricas de rede ou incremento de sua robustez.

II.1 REDES *Overlay*

Redes *overlay* não são uma idéia nova. A Internet iniciou sua vida como uma rede de dados sobreposta ao sistema de telefonia, e mesmo hoje um grande número de conexões da Internet continua sendo efetuada via linhas de telefone. Com o advento da rede mundial, aplicações comerciais [14], de entretenimento [10] e de pesquisa [2] foram desenvolvidas seguindo o modelo de redes *overlay*. Redes *overlay* também estão sendo utilizadas para possibilitar o uso de protocolos não suportados pelos roteadores da Internet atual, como a criação de redes IPv6 sobre rotas que suportam somente o IP versão 4. D. Andersen et al. utilizaram redes *overlay* sobre a Internet para identi-

ficar rotas alternativas, que usualmente não são utilizadas pelo protocolo de roteamento [2]. Estas rotas alternativas foram utilizadas para aumentar a taxa de entrega da rede e diminuir o tempo de recuperação de falhas. Tal abordagem foi utilizada como referência para este trabalho.

II.2 CONFIABILIDADE EM REDES DE SENSORES SEM FIO

Assim como as redes *overlay*, avanços em áreas da Gerência de Falhas, tais como tolerância a falhas e garantia de entrega de pacotes, têm sido feitos em RSSF. N. D. Georganas et al. utilizam um nó pivô para, a partir de informação prévia de roteamento existente em sua vizinhança, livrar-se de rotas quebradas [4]. Já Garurav Gupta et al. adotam um esquema de realocação de nós em novos *clusters* que, devido à falha de seus *cluster-heads* originais, tornaram-se inacessíveis [9]. Finalmente, Stefan Dulman et al., a fim de garantir que dados colhidos alcançarão seu destino, utilizam uma abordagem de escolha de caminhos baseada no histórico de falhas dos nós da rede [18].

II.3 REDES *Overlay* SOBRE REDES SEM FIO

Redes *overlay* e sem fio também já foram integradas. Chao Gui et al. utilizaram *Multicast* através de ROs em Redes Móveis para atingir robustez a baixo custo adicional [8]. Randy H. et al. e Marco Z. et al. defendem a idéia de que redes *overlay* são as soluções para integrar diferentes tecnologias sem fio de forma transparente e integrar redes baseadas em IP com RSSF [11, 21]. Além disso, Mark Stemm e Randy H. Katz lidaram com *handoffs* verticais em redes *overlay* sem fio [19]. Finalmente, L. B. Oliveira et al. avaliaram a utilização de redes *Peer-to-Peer* para compartilhamento de arquivos em MANETs [12].

III DESENHO E FUNCIONAMENTO

III.1 REDE *Overlay*

A rede *overlay* forma uma rede lógica sobre a rede física. Na rede lógica, nós exercem enlaces lógicos com os demais através de conexões virtuais. Estes, do ponto de vista da rede física subjacente, são caminhos *multi-hop* pelos quais o tráfego é roteado de forma transparente para a aplicação.

Neste trabalho a RO é constituída apenas pelos nós *cluster-head*, deixando aos sensores de menor poder energético e computacional o encargo de sensoriar a rede. Tal estratégia permite implementar a rede *overlay* a um baixo custo, uma vez que apenas um percentual baixo dos sensores possui *hardware* mais caro, que possui uma quantidade maior de energia armazenada na sua bateria.

Cada nó da rede *overlay* possui um grupo de vizinhos, que podem ser utilizados como intermediários em rotas alternativas à rota padrão – aquela determinada pelo protocolo de roteamento. O funcionamento de um nó da rede *overlay* pode ser visto na Figura 2.

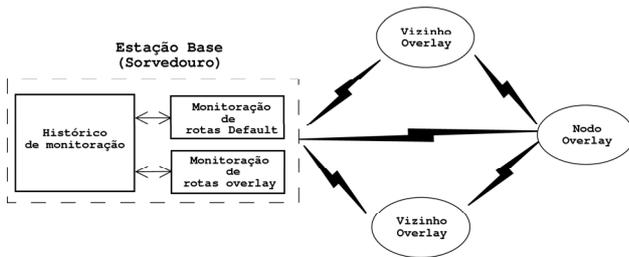


Figura 2: Diagrama da troca de mensagens na rede *overlay* simulada. As ligações entre nós mostram as rotas monitoradas pelo nó *overlay*.

III.2 MONITORAÇÃO DE ROTAS *Overlay*

Para que seja escolhida a melhor rota dentre as várias possíveis, é aferida a qualidade das mesmas através de monitoração. Podem ser obtidos dados como latência, *jitter*, número de *hops* e confiabilidade de cada rota. A partir destes, podemos determinar qual rota é a ideal para cada aplicação.

O monitoramento é feito através do envio periódico de pacotes dos nós *overlay* até a EB, através da rota padrão e de seus vizinhos *overlay*. A Estação Base fica então responsável por receber os dados de monitoração e os analisar, gerando estatísticas sobre as métricas de rede para cada rota monitorada. Feito isso, este dissemina as informações para os demais nós *overlay*, para que os mesmos as utilizem na escolha da melhor rota para a aplicação corrente.

III.3 AVALIAÇÃO E SELEÇÃO DE ROTA

A fim de avaliar qual das rotas é a mais adequada para a aplicação corrente, a Estação Base mantém um registro dos últimos n pacotes de monitoração enviados pelas rotas monitoradas. Como a granularidade do intervalo de monitoração é conhecida, a Estação Base pode inferir sobre a confiabilidade dos canais com base na taxa de chegada de pacotes de monitoração. A taxa de chegada de pacotes e o número de pacotes considerados na avaliação das métricas da rede influenciam na precisão dos resultados obtidos. Ao receber um pacote de monitoração, a EB determina outras métricas de rede, como *jitter* e latência.

Uma vez que a avaliação de rotas é realizada e seus resultados são disseminados pela EB, os nós *overlay* podem então optar entre as rotas *overlay* e as rotas padrão, verificando qual caminho atende melhor às necessidades da aplicação.

IV SIMULAÇÃO

Como não dispomos de um ambiente de Redes de Sensores Sem Fio real, fez-se necessário o uso de um simulador para a avaliação do esquema proposto. O simulador escolhido para tal foi o *Network Simulator (ns-2)* [17], versão 2.1b8a, amplamente utilizado pela comunidade de pesquisa em redes de computadores. Para realizar as simulações, desenvolvemos extensões para o simulador que modelam o funcionamento de uma rede *overlay*. Para avaliar o uso de uma rede *overlay*, simulamos uma rede

que utiliza a aplicação *overlay*, e outra que não utiliza a aplicação. Os cenários simulados serão detalhados a seguir.

IV.1 AMBIENTE SIMULADO

Para que avaliássemos os benefícios e os encargos que uma rede *overlay* proporciona, simulamos os seguintes cenários:

- Cenário 1: RSSF efetuando apenas a atividade de sensoriamento em uma topologia onde não há falha de nós.
- Cenário 2: RSSF efetuando apenas a atividade de sensoriamento em uma topologia onde 20% dos nós irão falhar durante a simulação.
- Cenário 3: RSSF efetuando apenas a atividade de sensoriamento em uma topologia onde 40% dos nós irão falhar durante a simulação.
- Cenário 4: RSSF utilizando uma RO, na qual além do sensoriamento eram realizadas monitoração, avaliação e escolha de rotas mais adequadas, em uma topologia onde não há falha de nós.
- Cenário 5: RSSF utilizando uma RO, na qual além do sensoriamento eram realizadas monitoração, avaliação e escolha de rotas mais adequadas, em uma topologia onde 20% dos nós irão falhar durante a simulação.
- Cenário 6: RSSF utilizando uma RO, na qual além do sensoriamento eram realizadas monitoração, avaliação e escolha de rotas mais adequadas, em uma topologia onde 40% dos nós irão falhar durante a simulação.

Para as situações acima, o ambiente simulado apresenta uma topologia hierárquica na qual sensores coletam dados periodicamente e os enviam para os nós *cluster-head*, que então encaminham os dados para a EB. A rede é heterogênea somente em relação a processamento e armazenamento de energia. Em outras palavras, os nós *cluster-head* possuíam um maior poder computacional e enérgico, mas uma vez que o alcance de transmissão era o mesmo dos nós comuns, seu envio de dados para a Estação Base era feito via caminhos *multi-hop*.

A partir do Cenário 4, além de exercer o sensoriamento do ambiente, os nós *cluster-head* participam de uma rede *overlay*, como descrita na Seção III.

Os nós *cluster-head* executavam a aplicação *overlay*, monitorando a rede à procura de rotas alternativas para transmissão. A Estação Base processava informações de monitoração e as repassava para o restante da rede *overlay* em um único *hop*. De posse das estatísticas de rotas, os nós *cluster-head* podiam, então, escolher os enlaces mais confiáveis para encaminhar os pacotes de dados.

Para que a monitoração fosse efetuada com o menor dispêndio de recursos, além de deixar a carga da EB todo o processamento, geração e disseminação de estatísticas,

assumimos que os nós *overlay* possuíam um segundo rádio, utilizado para receber as estatísticas de rotas – tal premissa é razoável, pois consideramos que os nós *cluster-head* possuem *hardware* mais poderoso. Desta forma, somente os nós *cluster-head* escutariam as informações de monitoração, poupando assim os demais nós do gasto de energia com o recebimento destas mensagens. Esta estratégia também possibilita o envio de informações de rotas para os nós *overlay* com maior frequência, diminuindo o tempo de resposta da rede em situações de falhas de nós.

IV.2 CONFIGURAÇÃO

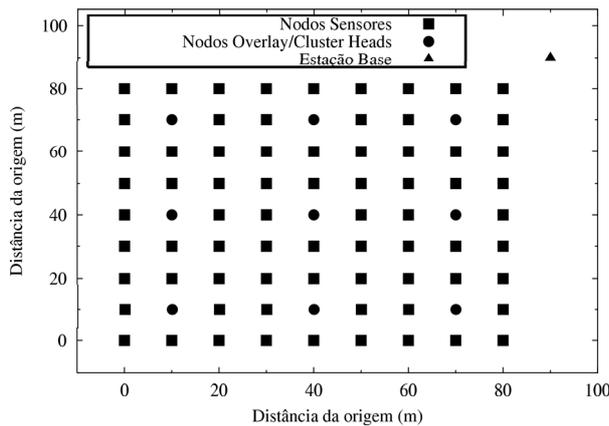


Figura 3: Topologia da rede simulada. Para a topologia mostrada, foram formados 9 grupos de nós.

Os experimentos simulados foram feitos em uma rede composta de 9 nós *cluster-head* e 72 nós sensores comuns. Os nós estavam dispostos em um *grid* 90×90 metros, no qual o *cluster-head* desempenha o papel de nó *overlay*. Os grupos possuíam 8 nós sensores, que se localizavam em volta do *cluster-head*. A distância entre os nós *overlay* era de 30 metros, enquanto a distância entre os nós sensores era de 10 metros. A Estação Base estava localizada a 10 metros acima e à direita do nó mais acima e à direita da rede. A disposição dos nós pode ser visualizada na Figura 3. Foi utilizado como protocolo de camada MAC o IEEE 802.11b e para roteamento o protocolo DSR [20].

Para que os nós ainda possuísem energia ao final da simulação, atribuímos aos mesmos uma alta energia inicial, para que as falhas ocorressem somente devido ao nosso modelo de falhas.

Definimos a energia usada para transmissão igual a 0.3 *Watts*, enquanto a energia gasta na recepção era de 0.2 *Watts*. Estes dados foram calibrados levando em consideração rádios existentes para RSSF [5]. As simulações feitas registraram o tráfego da rede durante um período de 100 segundos.

Um percentual fixo de nós, escolhidos aleatoriamente, irá falhar de acordo com uma distribuição uniforme no tempo. No modelo de falhas utilizado, o nó não irá enviar ou receber dados até o fim da simulação, após ocorrida uma falha. Todos os nós, excetuando-se a EB, podem falhar.

Assim como feito por D. Andersen em [2], optamos por ter dois vizinhos para cada nó *overlay*. Estes vizinhos são

escolhidos ao início da simulação de forma aleatória entre os nós *overlay*, e são mantidos fixos até o final da mesma. Estipulamos que a Estação Base envia dados de atualização de melhores rotas para os nós *overlay* a cada 1.5 segundos.

Os parâmetros de simulação se encontram sumarizados na Tabela 1.

Tabela 1: Parâmetros de Simulação.

Parâmetro	Valor
Dimensão do <i>Grid</i>	90 x 90m
Número de sensores comuns	72
Número de <i>cluster-heads</i>	9
Distância entre nós	10m
Alcance do rádio	15m
Número de vizinhos <i>overlay</i>	2
Intervalo de sensoriamento	1.5s
Intervalo de monitoração	1.5s
Intervalo de disseminação de rotas	1.5s
Energia inicial	100J
Algoritmo de roteamento	DSR
Camada MAC	IEEE 802.11b
Tempo de simulação	100s

IV.3 MÉTRICAS DE AVALIAÇÃO

Com o intuito de avaliar o comportamento de uma RO sobre RSSF, foram levados em conta parâmetros de rede comumente utilizados, porém ressaltando aqueles cujos valores são mais importantes em ambientes com escassez de recursos computacionais – tal como acontece em RSSF. São eles:

- Taxa de entrega média.
- Energia média gasta por *cluster-head*.
- Energia média gasta por nó comum.
- Energia média gasta por nós em geral.
- *Jitter* médio.
- Latência média.
- *Hops* médios.

O comportamento de cada uma dessas métricas foi avaliado para os 8 cenários. Os resultados são apresentados na próxima Seção.

V RESULTADOS

A partir do comportamento das redes de sensores de todos os cenários listados em IV.2, e levando em consideração as métricas descritas em IV.3, foram obtidos os gráficos apresentados nesta seção.

O gráfico na Figura 4 apresenta a taxa de entrega para os cenários simulados. As duas curvas dos gráficos correspondem aos cenários que instanciam ou não instanciam a rede *overlay*. No eixo *X* está representado o percentual de nós que falharam durante a simulação, e no eixo *Y* é

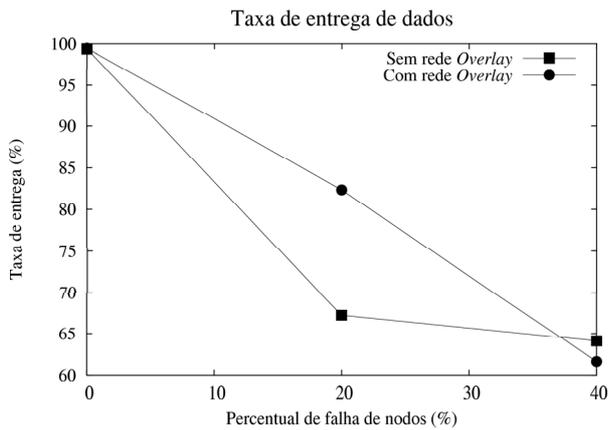


Figura 4: Taxa de entrega média.

mostrada a taxa de entrega da aplicação de sensoriamento *Sensor Comum* ⇒ *Estação Rádio Base* para os cenários simulados.

Pode ser observado que quando não há falha de nós a rota padrão é capaz de entregar dados para a EB de maneira satisfatória. Com isso, a monitoração da rede à procura de rotas alternativas mais confiáveis é desnecessária. Por outro lado, na situação em que 20% dos nós da rede falham, o esquema provido pela RO é mais eficiente que o protocolo de roteamento, pois os protocolos de roteamento arquitetados para este tipo de rede, por serem projetados para escalabilidade e economia de recursos, restringem as possibilidades de rotas. Finalmente, quando é introduzida uma alta taxa de falhas, a taxa de entrega volta a se igualar à de um cenário não monitorado. Isto porque, diante de um alto índice de falhas mesmo rotas alternativas são quebradas.

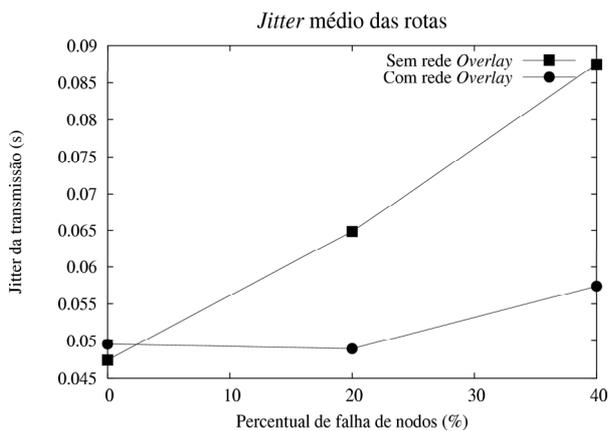


Figura 5: Jitter médio da rede.

Assim como a taxa de entrega, *jitter* e latência médios foram melhorados com a introdução da rede *overlay* em um cenário com falhas. Pela Figura 5, observa-se que em um cenário em que não ocorrem falhas a rede sem a aplicação *overlay* apresenta resultado próximo ao do obtido por uma rede com *overlay* – cerca de 0.02s mais baixo. Contudo, com a introdução de 20% e 40% de falhas na rede, o es-

quema, além de tornar o comportamento da curva mais homogêneo, resulta respectivamente em melhoras de aproximadamente 15 e 34 milissegundos. Neste caso a melhora perdura mesmo quando o percentual de falhas é incrementado, porque estas métricas são computadas somente a partir dos pacotes que conseguem alcançar a EB. Em outras palavras, o aumento das falhas acarreta em uma diminuição de rotas para a Estação Base, que por sua vez resulta em uma menor taxa de entrega para a rede com e sem o recurso *overlay*. Todavia, no cenário de 40% de falhas, apesar da RO não conseguir manter um alto índice de entrega, os pacotes que são entregues possuem menor *jitter* médio.

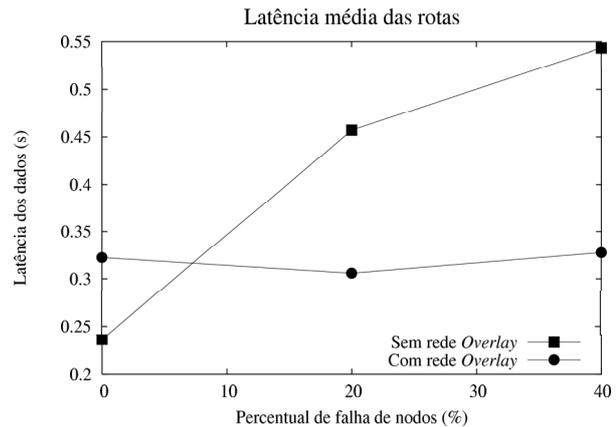


Figura 6: Latência média da rede.

A estratégia *overlay* obteve bons resultados em relação ao atraso médio de pacotes pelas mesmas razões identificadas para o *jitter* médio. Como pode ser observado no gráfico da Figura 6, os percentuais de melhora para 20% e 40% de falhas quando se compara a RSSF com e sem RO são de aproximadamente 33% e 40%, respectivamente. Note, porém, que ao considerarmos o ambiente sem falhas (percentual de falhas igual a zero), a latência média do cenário sem *overlay* é cerca de 28% menor. Este resultado, que era esperado, é devido ao maior caminho que o pacote de dados percorre em direção à EB quando a rede *overlay* é instanciada – neste cenário, o pacote é encaminhado para um nó *overlay* intermediário, para então seguir para a Estação Base.

Visto os benefícios atingidos com a escolha de melhores rotas, apresentaremos a seguir gráficos de pacotes enviados e energia média consumida. O objetivo é avaliar a viabilidade do esquema em função de seus custos e das demandas de possíveis aplicações.

A Figura 7 apresenta a quantidade de pacotes enviados, respectivamente, em uma rede *overlay* e em uma rede sem *overlay*, ambas sem falhas. A rede *overlay* enviou 39% mais pacotes que a mesma sem a aplicação rodando. Esta diferença é devida à monitoração da rede. Os nós *overlay* devem monitorar rotas que podem ter uma confiabilidade muito baixa, e assim possuem grande taxa de perda.

Os gráficos das Figuras 8, 9 e 10 apresentam, respectivamente, o consumo de energia dos nós *cluster-head*, dos nós comuns e a média dos mesmos para o ambiente ideal. Como esperado, o maior gasto de energia ficou concen-

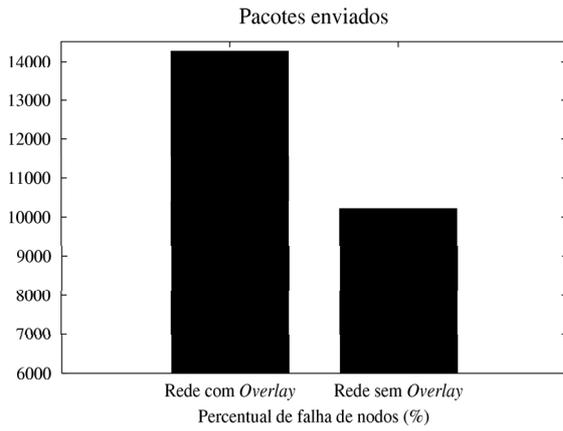


Figura 7: Pacotes enviados, para uma rede sem falhas.

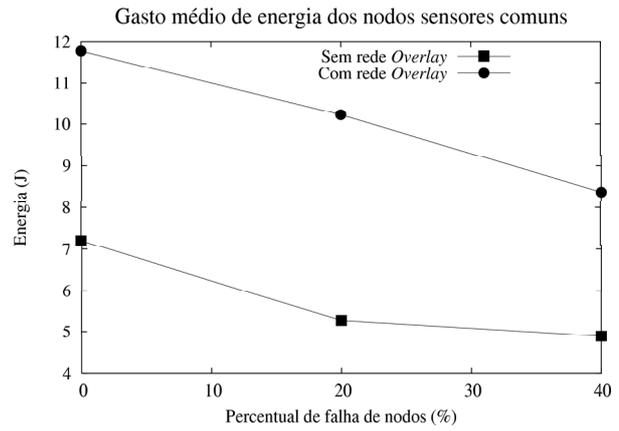


Figura 9: Gasto médio de energia dos nós sensores comuns.

trado nos nós *cluster-head* – 21.8 e 17.6J para os cenários de falhas, com e sem o esquema *overlay*, respectivamente. Além disso, a introdução da rede *overlay* resultou em um gasto médio de energia 26% maior. Tal aumento ocorre devido ao envio dos pacotes de monitoração da rede, e ao maior tamanho das rotas utilizadas para transmissão de dados (o número de *hops* médios das rotas passou de 8.5 para 10 com a introdução da aplicação *overlay*). Por isso, os nós da rede gastam mais energia roteando pacotes. Pode-se constatar pelos gráficos que quanto maior o percentual de falha de nós na rede, menor é o gasto de energia dos mesmos, pois os nós que desempenham o papel de roteadores devem rotear uma menor quantidade de dados.

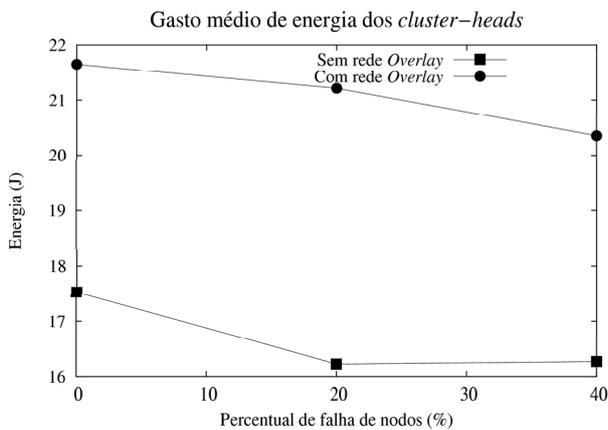


Figura 8: Gasto médio de energia dos nós *cluster-heads*.

VI CONCLUSÃO E TRABALHOS FUTUROS

Este trabalho apresentou, através de simulações, uma avaliação de uma Rede de Sensores Sem Fio que utiliza uma rede *overlay* no intuito de aumentar a confiabilidade da transmissão de dados.

Pelo fato de protocolos de roteamento para redes *ad hoc* em geral terem sido projetados para serem escaláveis e consumirem poucos recursos, sejam computacionais ou energéticos, caminhos alternativos são desconsiderados. A estratégia proposta neste trabalho possibilitou incrementar a

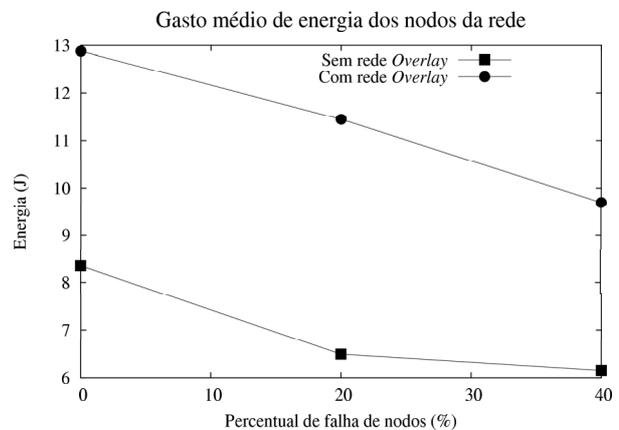


Figura 10: Gasto médio de energia dos nós.

qualidade de métricas de rede como confiabilidade, latência e *jitter* médios, ao considerar estes caminhos alternativos. Foi verificado que o consumo de energia devido à utilização de redes *overlay*, apesar de alto, pode ser justificado caso a rede necessite de altas taxas de entrega.

Como trabalhos futuros, pretende-se explorar ainda mais as possibilidades que uma rede *overlay* ocasiona. Para tal, um estudo de como escolher vizinhos para os nós *overlay*, levando em consideração suas posições no *grid*, bem como o número ideal de vizinhos, pode aprimorar os resultados obtidos. Além disso, existe a possibilidade de aumentar o número de *hops overlay* que um pacote de dados percorre em direção à Estação Base. Finalmente, pode-se efetuar a escolha de rotas baseadas em métricas específicas de RSSF, como por exemplo, rotas de menor consumo computacional ou energético.

AGRADECIMENTOS

Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), uma entidade do governo brasileiro voltada ao desenvolvimento científico e tecnológico, processo número 55.2111/2002-3, projeto SensorNet.

Referências

- [1] I. F. Akyldiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Computer Networks*, pages 393–422, December 2002.
- [2] David Andersen, Hari Balakrishnan, Frans Kaashoek, and Robert Morris. Resilient overlay networks. In *Proceedings of the eighteenth ACM symposium on Operating systems principles*, pages 131–145. ACM Press, 2001.
- [3] C. M. Cordeiro and D. P. Agrawal. Minicurso: Redes móveis ad hoc. In *20º Simpósio Brasileiro de Redes de Computadores*, Maio 2002.
- [4] Nicolas D. Georganas Di Tia. Energy efficient routing with guaranteed delivery in wireless sensor networks. In *Proceedings of the Wireless Communications and Networking Conference 2003 (WCNC)*, New Orleans, USA, March 2003. IEEE.
- [5] Linnyer B. Ruiz Fabrício A. Silva, Thais R. de Moura Braga. Tecnologia de nodos de sensores sem fio, 2002.
- [6] Freenet. <http://www.freenet.com>.
- [7] Gnutella. <http://www.gnutella.com>.
- [8] Chao Gui and Prasant Mohapatra. Efficient overlay multicast for mobile ad hoc networks. In *Proceedings of the Wireless Communications and Networking Conference 2003 (WCNC)*, New Orleans, USA, March 2003. IEEE.
- [9] Gaurav Gupta and Mohamed Younis. Fault-tolerant clustering of wireless sensor networks. In *Proceedings of the Wireless Communications and Networking Conference 2003 (WCNC)*, New Orleans, USA, March 2003. IEEE.
- [10] Icq. <http://web.icq.com>.
- [11] Randy H. Katz and Eric A. Brewer. The case for wireless overlay networks. In Tomasz Imielinski and Henry F. Korth, editors, *Mobile Computing*, pages 621–650. Kluwer Academic Publishers, Computer Science Division, Department of Electrical Engineering and Computer Science, University of California, Berkeley, 1996.
- [12] Leonardo B. Oliveira, Isabela Siqueira, Antonio A. Loureiro. Evaluation of ad hoc routing protocols under a peer-to-peer application. In *IEEE Wireless Communications and Network Conference*, New Orleans, USA, march 2003. IEEE.
- [13] A. A. Loureiro, Linnyer B. Ruiz, Raquel A. Mini, and José Marcos S. Nogueira. Rede de sensores sem fio. *Simpósio Brasileiro de Computação - Jornada de Atualização de Informática*. Florianópolis, SC, Brasil, 2002.
- [14] Gang Peng. CDN: Content distribution network. Technical Report TR-125, Experimental Computer Systems Lab, 2003.
- [15] Praveen Ravi Udit Shashi. Survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, 2002.
- [16] S. Shi and J. Turner. Routing in overlay multicast networks. In *IEEE Infocom 2002*, New York, USA, June 2002. IEEE.
- [17] NS 2 Simulator. Disponível via <http://www.isi.edu/nsnam/ns/>. Acessado em 10 de abril de 2002.
- [18] Jian Wu Paul Havinga Stefan Dulman, Tim Nieberg. Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks. In *Proceedings of the Wireless Communications and Networking Conference 2003 (WCNC)*, New Orleans, USA, March 2003. IEEE.
- [19] Mark Stemm and Randy H. Katz. Vertical handoffs in wireless overlay networks. *Mobile Networks and Applications*, 3(4):335–350, 1998.
- [20] V. D. Park and M. S. Corson. A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks. In *INFOCOM (3)*, pages 1405–1413, 1997.
- [21] Marco Z. and Bhaskar Krishnamachari. Integrating future large-scale wireless sensor networks with the internet. Technical Report CS 03-792, Department of Electrical Engineering, University of Southern California, 2003.

Uma Análise do Consumo de Energia em Redes Ad Hoc

Daniel de O. Cunha, Luís Henrique M. K. Costa e Otto Carlos M. B. Duarte

Grupo de Teleinformática e Automação
PEE/COPPE – DEL/POLI
Universidade Federal do Rio de Janeiro
<http://www.gta.ufrj.br/>

Abstract – This paper analyzes the energy consumption of ad hoc nodes using IEEE 802.11 interfaces. Our objective is to provide theoretical limits on the lifetime gains that can be achieved by different power saving techniques found in the literature. The evaluation takes into account the properties of the medium access control protocol and the process of forwarding packets in ad hoc mode. The key idea is to determine the node lifetime based on its average power consumption. The average power consumption is estimated considering how long the node remains idle, receiving, or transmitting. We show that energy-aware routing can achieve a lifetime gain as large as 30%, and significant gains can be obtained even if a small number of paths is used.

Keywords – Energy Consumption, Wireless Communication, Ad Hoc Networks.

Resumo – Este artigo analisa o consumo de energia dos nós em redes ad hoc IEEE 802.11. O principal objetivo desta análise é descobrir limites teóricos para os ganhos obtidos em tempo de vida dos nós através de diferentes técnicas de conservação de energia encontradas na literatura. A avaliação do consumo de energia considera as características do método de acesso ao meio e a interação dos nós no processo de encaminhamento dos pacotes. A idéia básica é determinar o tempo de vida de um nó a partir do seu consumo médio. O consumo é estimado a partir da parcela de tempo que o nó permanece transmitindo, recebendo ou ocioso. São então analisados os ganhos potenciais obtidos por diferentes técnicas de economia ou de balanceamento do consumo de energia. Mostra-se que a técnica de balanceamento de fluxos pode alcançar um ganho máximo para o tempo de vida dos nós entre 11% e 30%, dependendo da configuração da rede.

Palavras-Chave – Consumo de Energia, Comunicação sem Fio, Redes Ad Hoc.

I. INTRODUÇÃO

Um fator crítico na operação das redes ad hoc é o consumo de energia dos nós. Tipicamente, nós ad hoc são dispositivos sem fio, alimentados por baterias cujas capacidades de armazenamento são limitadas por restrições de volume e peso dos equipamentos. Por isto, é importante reduzir o consumo de energia dos nós. Além disso, as redes ad hoc funcionam sem nenhuma infraestrutura pré-estabelecida, obrigando os nós a cooperar

entre si a fim de viabilizar comunicações entre nós distantes. Assim, a desativação de um nó por falta de energia pode prejudicar o funcionamento de toda a rede. Desta forma, a economia de energia é importante para um nó, isoladamente, mas também para a rede como um todo.

O problema da conservação de energia em redes ad hoc vem sendo abordado de diversas formas. Uma delas é utilizar a energia restante nos nós como métrica de roteamento, técnica que foi discutida inicialmente por Singh *et al.* [1]. O objetivo da energia como métrica de roteamento é evitar a utilização de uma mesma rota repetidas vezes, o que levaria ao rápido esgotamento da energia dos nós presentes nesta rota. Este tipo de roteamento procura balancear os fluxos de dados por diferentes rotas, levando em consideração a energia restante nos nós da rede. O balanceamento de tráfego é mais simples em redes estáticas onde a topologia pode ser seguramente obtida, porém é mais complexo em redes móveis que possuem topologias dinâmicas. Chang e Tassiulas [2] apresentam algoritmos para balancear fluxos com o objetivo de maximizar o tempo de vida dos nós da rede. Yu e Lee [3] propõem duas novas versões do protocolo *Dynamic Source Routing* [4] que visam maximizar o tempo até que o primeiro nó da rede fique sem energia. Visando reduzir o atraso na descoberta de rotas, Cunha *et al.* [5] propõem modificações a um dos mecanismos de Yu e Lee, bem como estendem seu uso a protocolos reativos em geral. Kim e Cho [6] desenvolveram um algoritmo de roteamento capaz de alterar as rotas durante as conexões, protegendo nós próximos à exaustão. Outros trabalhos assumem que, devido à atenuação exponencial de sinais na transmissão sem fio, a comunicação através de pequenos e múltiplos saltos é mais eficiente que a transmissão com menos saltos, porém maior distância entre os nós. Wan *et al.* [7] analisam a melhor rota numa determinada topologia considerando a potência de transmissão dos nós variável. A abordagem de Singh *et al.* [8] busca o balanceamento no consumo de energia dos nós dividindo as conexões em dois segmentos, um de comunicação com múltiplos saltos e outro com comunicação direta.

A maior parte dos trabalhos considera a utilização de interfaces sem fio IEEE 802.11 na construção de redes ad hoc. No entanto, estas interfaces no modo ad hoc possuem características específicas de operação, que muitas vezes não têm sido levadas em consideração. Feeney e Nilsson [9] apresentam medidas com interfaces IEEE 802.11 e modelam a energia consumida por pacote em relação ao consumo da interface no estado ocioso. No mesmo trabalho é mostrado que o custo deste estado ocioso é

relativamente alto, pois os nós devem ficar constantemente escutando o meio para receber transmissões de seus vizinhos. Monks *et al.* [10] analisam os efeitos da utilização de diferentes potências de transmissão no consumo de energia dos nós.

Neste trabalho é analisado o consumo de energia dos nós ad hoc levando-se em consideração a interação entre o método de acesso ao meio de interfaces IEEE 802.11 e o encaminhamento de pacotes em modo ad hoc. Para isto, são levadas em consideração as parcelas de tempo em que os nós permanecem num dado estado de operação e a capacidade de transmissão de dados de redes ad hoc. Por fim, este trabalho demonstra como os resultados podem ser utilizados para investigar o ganho potencial de diversas técnicas de economia ou balanceamento de energia. Estes ganhos limites podem ser utilizados como referência na pesquisa de novos esquemas de economia de energia.

Este trabalho está organizado da seguinte forma. A Seção II descreve as características principais da operação das redes IEEE 802.11 no modo ad hoc, destacando os estados possíveis de funcionamento das interfaces que seguem este padrão. Na Seção III é analisada a interação entre o encaminhamento em modo ad hoc e o consumo de energia dos nós. A Seção IV discute como os resultados apresentados podem ser utilizados para calcular limites possíveis para o ganho de diversas técnicas de economia ou de balanceamento de energia. Por fim, A Seção V apresenta as conclusões deste trabalho.

II. REDES IEEE 802.11

O padrão IEEE 802.11 [11] é a tecnologia de redes locais sem fio mais utilizada atualmente. O IEEE 802.11 possui dois mecanismos de acesso ao meio. O esquema distribuído de acesso ao meio (*Distributed Coordination Function - DCF*) é utilizado no modo de operação ad hoc enquanto que o esquema centralizado (*Point Coordination Function - PCF*) é usado em redes infra-estruturadas. A seção a seguir resume o DCF, foco deste trabalho.

II.1. DISTRIBUTED COORDINATION FUNCTION - DCF

O DCF utiliza o método de acesso *Carrier-Sense Multiple Access with Collision Avoidance (CSMA/CA)*. Para obter controle do meio e transmitir um quadro de dados, o terminal deve primeiro escutar o meio por um certo tempo (DIFS - *DCF InterFrame Space*) e se o meio continuar livre ele envia o quadro, como mostra a Figura 1. Devido à diferença de potência entre o sinal transmitido e o recebido, o emissor não pode detectar se o quadro foi corretamente recebido, tornando necessário o envio de um quadro de reconhecimento (ACK) pelo receptor. Após receber o quadro de dados, o receptor aguarda um período SIFS (*Short InterFrame Space*) e transmite o ACK para o emissor do quadro. O fato de SIFS ser menor que DIFS garante que o receptor terá oportunidade de enviar seu reconhecimento antes que algum outro nó ocupe o meio com outro quadro de dados. Para garantir acesso justo ao meio e reduzir as chances de colisões, nós que desejem enviar um segundo pacote deverão escutar o meio por um

período igual ao DIFS somado a um tempo aleatório (*backoff*).

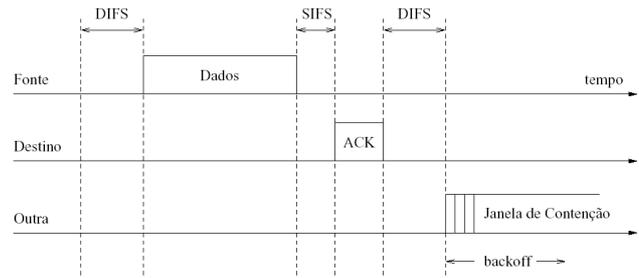


Fig. 1. Esquema básico do DCF.

O DCF implementa, ainda, um mecanismo virtual de escuta ao meio chamado *Network Allocation Vector (NAV)* que guarda por quanto tempo o meio ficará ocupado. Para viabilizar este mecanismo, existem dois outros tipos de quadros de controle na camada MAC 802.11, o *Request to Send (RTS)* e o *Clear to Send (CTS)*. Este mecanismo visa reduzir colisões e evitar o problema do terminal escondido nestas redes. Este problema surge quando um nó vizinho do receptor de uma comunicação não consegue detectar uma transmissão em andamento, por estar muito longe do emissor, e decide iniciar uma outra comunicação, causando colisão no receptor da primeira comunicação. Assim, cada nó que deseja transmitir um quadro de dados primeiro envia um quadro de pedido RTS que informa a duração calculada para a comunicação. Caso o nó destino receba corretamente o RTS ele envia em resposta um quadro CTS. O quadro CTS enviado pelo receptor do quadro de dados contém a duração estimada para o resto da comunicação. Todas as estações vizinhas que não participam diretamente da comunicação, mas que podem escutar o tráfego, chamadas de estações ouvintes, atualizam seus vetores de alocação (NAVs). Para que este mecanismo funcione, todos os nós ouvintes devem ser capazes de receber as informações sobre a duração da comunicação. Por isto, os quadros RTS e CTS devem ser transmitidos numa das taxas básicas suportadas por todos os nós da rede, independentemente da taxa máxima de transmissão de dados de cada nó. Tipicamente, a taxa utilizada para a negociação com RTS e CTS é de 1Mbps. Esta negociação pode ser vista na Figura 2.

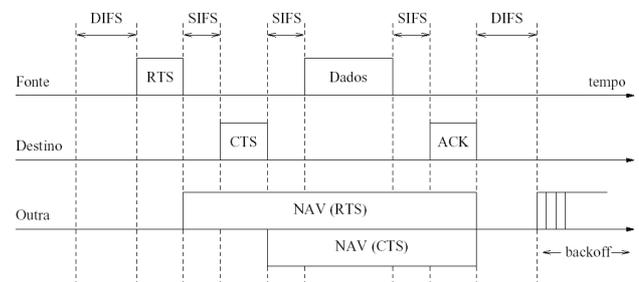


Fig. 2. DCF com negociação por RTS/CTS.

II.2. ESTADOS DE OPERAÇÃO DA INTERFACE

As interfaces IEEE 802.11 possuem 4 estados de operação bem definidos: o estado de dormência (*Sleep*), o estado ocioso (*Idle*), o estado de transmissão (Tx) e o

estado de recepção (Rx). Dentre estes estados o que apresenta o menor consumo de energia é o estado de dormência, porém os nós não podem ficar neste estado todo o tempo que não estiverem efetivamente recebendo ou transmitindo já que neste estado não podem detectar transmissões das quais sejam destinatários nem iniciar imediatamente transmissões de pacotes. Este comportamento pode ser visto na Figura 3.

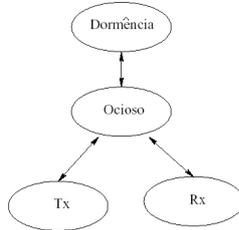


Fig. 3. Estados das interfaces sem fio.

A tarefa de gerenciar a entrada e saída dos nós do estado de dormência é particularmente complexa em redes ad hoc uma vez que não há nenhum elemento centralizador. Além disso, seria necessário que nós vizinhos armazenassem pacotes destinados a nós que estejam em seus períodos de dormência. Por estes motivos costuma-se considerar que os nós ad hoc ficam no estado ocioso quando não estão transmitindo ou recebendo dados, não utilizando o estado de dormência. O estado ocioso apresenta um consumo mais alto que o consumo no estado de dormência. Isto ocorre porque no modo ad hoc os nós devem ficar constantemente escutando o meio para saber se há algum quadro endereçado a eles. O estado Rx apresenta um consumo, em geral, pouco maior que o consumo no estado ocioso. Este aumento representa o custo de efetivamente processar os sinais recebidos ao invés de apenas detectar a presença ou não de um sinal. Dentre todos os estados, o estado Tx é o que apresenta o maior consumo. O principal motivo é a grande atenuação de espaço livre que força a inclusão de amplificadores de potência para RF nestas interfaces. Algumas interfaces possibilitam a mudança da potência de saída do sinal, reduzindo assim o consumo de energia como um todo.

III. CONSUMO DE ENERGIA DOS NÓS

As análises realizadas nesta seção assumem o uso de interfaces IEEE 802.11b com taxa máxima de transmissão de 11Mbps operando no modo DCF e utilizando a negociação por RTS/CTS para reduzir colisões. Considerando os quatro estados possíveis das interfaces, pode-se obter potências médias (P_m) que representem a energia consumida pela interface. A potência média pode ser obtida da seguinte forma:

$$P_m = t_d \times P_d + t_{oc} \times P_{oc} + t_{RX} \times P_{RX} + t_{TX} \times P_{TX}, \quad (1)$$

onde, t_d , t_{oc} , t_{RX} e t_{TX} são as frações de tempo que o nó fica em cada um dos estados possíveis: dormência, ocioso, Rx e Tx, respectivamente. Estas frações de tempo devem satisfazer a condição $t_d + t_{oc} + t_{RX} + t_{TX} = 1$.

Analogamente, P_d , P_{oc} , P_{RX} e P_{TX} são as potências instantâneas consumidas em cada um dos quatro estados. Com base em P_m e na energia inicial do nó (E), pode-se

calcular o tempo de vida do nó (T_v), que representa o tempo até que a energia do nó chegue a zero, como:

$$T_v = \frac{E}{P_m} \quad (2)$$

A análise do tempo de vida apresentada neste trabalho leva em conta apenas a energia consumida pelas interfaces sem fio dos nós, desconsiderando o consumo de outros circuitos dos equipamentos.

Esta análise assume, ainda, a inexistência de qualquer técnica que possibilite a transição para o estado de dormência, ou seja, $t_d = 0$. Com esta restrição, o tempo de vida máximo de um nó pode ser calculado considerando-se que este nó permanece no estado ocioso todo o tempo (T_{oc}), como mostrado na Equação 3.

$$T_{oc} = \frac{E}{P_{oc}} \quad (3)$$

Inicialmente, pode-se intuir que o menor tempo de vida possível de um nó seria no caso em que ele passasse todo o tempo transmitindo. No entanto, esta hipótese não é real. Mesmo que um nó fique continuamente enviando quadros de dados, ele passará alguns momentos no estado ocioso e no estado Rx pelo próprio funcionamento do mecanismo de acesso ao meio (Seção II.1). Para verificar os efeitos do esquema DCF sobre o consumo de energia dos nós, a primeira análise realizada é num cenário de comunicação direta entre dois nós, que possibilita uma capacidade de transmissão máxima, já que não há contenção. Após esta análise, será feita uma análise dos efeitos do encaminhamento de quadros em modo ad hoc.

III.1. COMUNICAÇÃO DIRETA

Esta análise é importante pois serve como base para análises mais complexas. O cenário considerado é de dois nós separados por uma distância que permita a comunicação direta. O nó emissor tenta transmitir dados a 11Mbps, garantindo que sempre que o meio esteja livre ele tenha algo a transmitir. Desta forma, a taxa máxima de utilização da rede pode ser calculada. Nesta condição, t_{oc} , t_{TX} e t_{RX} são a fração do tempo da transmissão de um quadro que os nós passam nos estados ocioso, Tx e Rx, respectivamente. Desprezando-se os atrasos de propagação, as parcelas que compõem o tempo de transmissão de um quadro podem ser vistas na Figura 4.

O tempo de *backoff* neste cenário pode ser calculado considerando-se uma janela de contenção igual a 31 (CW_{min}) todo o tempo. O *backoff* será, então, uniformemente distribuído entre 0 e 31 *slots* de tempo com duração de 20 μ s. A média do *backoff* será de 15,5 *slots* de tempo, resultando num tempo médio de 310 μ s por quadro. A norma define o SIFS como 10 μ s e o DIFS como SIFS + 2 x *slot*, resultando em DIFS = 50 μ s. Além disso, antes de cada quadro MAC deve ser enviado um preâmbulo da camada física, que pode ser longo, de 192 μ s, ou curto, de 96 μ s [12]. Como o preâmbulo curto não é compatível com dispositivos mais antigos, será utilizado o preâmbulo longo nos cálculos desta seção. Os quadros de controle RTS, CTS e ACK são transmitidos na maior taxa básica possível, neste caso, será considerada 1Mbps. Assim a transmissão do RTS, com 20 *bytes*, leva 160 μ s, enquanto que a transmissão dos 14 *bytes* que

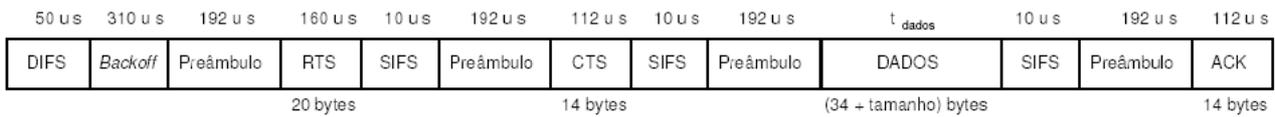


Fig. 4. Composição do tempo de transmissão de um quadro completo.

TABELA I
VALOR DE t_{dados} PARA DIFERENTES TAMANHOS DE PACOTE.

Tamanho do Pacote (bytes)	t_{dados} (μs)	Tempo do Quadro (μs)	Número de Quadros por seg.
160	156	1698	589
512	412	1954	512
800	621	2163	462
1000	747	2309	434
1500	1130	2672	374

compõem o CTS e o ACK leva 112 μs . O tempo de transmissão do quadro de dados é igual ao tempo de transmissão a 11Mbps de 34 bytes do cabeçalho MAC mais os bytes de dados. É importante ressaltar que nos bytes de dados devem ser contados quaisquer cabeçalhos de camadas superiores que venham a ser utilizadas. O cabeçalho IP, por exemplo, possui 20 bytes.

O tempo de transmissão de um quadro completo pode ser calculado da seguinte forma:

$$T_{quadro} = t_{Backoff} + 4 \times t_{pr} + 3 \times SIFS + DIFS + t_{RTS} + t_{CTS} + t_{dados} + t_{ACK} \quad (4)$$

Assim, o tempo de transmissão de um quadro completo, em μs , é igual a:

$$T_{quadro} = 1542 + t_{dados} \quad (5)$$

A Tabela I mostra os tempos t_{dados} e o número de quadros completos que podem ser transmitidos por segundo para diferentes tamanhos de pacote utilizados para os dados do usuário. Estes tempos foram calculados considerando-se um cabeçalho de 20 bytes (igual ao cabeçalho IP) acrescido a estes dados.

A partir destes diferentes t_{dados} e tempos de quadros pode-se calcular t_{oc} , t_{Tx} e t_{Rx} para o nó emissor e para o nó receptor. Os tempos de backoff, DIFS e SIFS são tempos em que ambos os nós ficam no estado ocioso. Os tempos referentes ao RTS e quadro de dados, incluindo os preâmbulos associados, são tempos em que o emissor fica no estado Tx e o Receptor no estado Rx. Nos períodos relativos ao CTS e ao ACK, a situação é inversa. Os valores calculados podem ser vistos na Tabela II.

TABELA II

PARCELA DE TEMPO QUE OS NÓS PASSAM EM CADA ESTADO.

Tamanho do Pacote (bytes)	Emissor			Receptor		
	t_{oc}	t_{Tx}	t_{Rx}	t_{oc}	t_{Tx}	t_{Rx}
160	0,23	0,41	0,36	0,23	0,36	0,41
512	0,20	0,49	0,31	0,20	0,31	0,49
800	0,18	0,54	0,28	0,18	0,28	0,54
1000	0,17	0,57	0,26	0,17	0,26	0,57
1500	0,14	0,63	0,23	0,14	0,23	0,63

Com base nestes valores pode-se calcular diferentes P_m segundo a Equação 1, um para cada tamanho de pacote. Assim, o uso de pacotes de 1500 bytes, por

exemplo, resulta em $P_m = 0,14 \times P_{oc} + 0,63 \times P_{Tx} + 0,23 \times P_{Rx}$ para o emissor e $P_m = 0,14 \times P_{oc} + 0,23 \times P_{Tx} + 0,63 \times P_{Rx}$ para o nó receptor.

Para verificar estes cálculos foram realizadas simulações no simulador *Network Simulator (ns-2)* [13]. Os cenários eram compostos por 2 nós distantes 200m um do outro, permitindo comunicação direta com o alcance de 250m do rádio. Os nós eram equipados com interfaces IEEE 802.11 operando a 11Mbps. O nó emissor tenta enviar dados a 11Mbps de modo a que sempre haja um pacote a ser transmitido quando o meio estiver livre. As simulações realizadas tiveram duração de 300s, e foi medido o tempo em que os nós ficaram em cada estado de operação. Foram calculados intervalos de confiança de 99% em relação à média das medidas, ficando estes intervalos menores que 0,5% desta média. A Figura 5 mostra os resultados obtidos para o nó emissor.

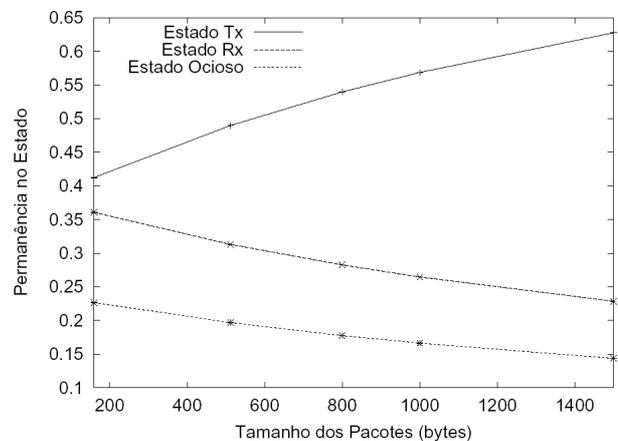


Fig. 5. Permanência do emissor nos estados de operação.

Como se pode observar na Figura 5, os resultados são bem próximos dos calculados. Além disto, podemos perceber que quanto maior o pacote utilizado, mais tempo a interface passa no estado Tx e menos tempo ela passa nos estados Rx e Ocioso. Isto implica numa maior potência média consumida e, conseqüentemente, num menor tempo de vida. No entanto, este menor tempo de vida não significa que utilizando pacotes menores a eficiência em termos de dados transmitidos seja maior. Como se pode ver na Tabela III, o uso de pacotes menores reduz a taxa máxima de envio de dados possível. Desta forma, para que a utilização de pacotes menores fosse mais vantajosa, seria necessário que a diferença de tempo

de vida do nó fosse grande a ponto de compensar esta diferença na taxa de envio.

TABELA III
TAXA MÁXIMA DE ENVIO DE DADOS.

Tamanho do Pacote (bytes)	Taxa Máxima de Envio (Mbps)
160	0,75
512	2,1
800	2,95
1000	3,47
1500	4,49

Considerando os valores da Tabela II podemos construir tabelas de permanência nos estados para nós ouvintes, ou seja, aqueles que não participam diretamente da comunicação mas estão no alcance do emissor e/ou receptor. Como mostra a Figura 6, existem três tipos de ouvintes: o ouvinte_e, que está no alcance apenas do emissor e, portanto, só escuta o tráfego originado pelo emissor; o ouvinte_r, que está no alcance apenas do receptor e, portanto, só escuta o tráfego originado pelo receptor; e o ouvinte_{er} que está no alcance de ambos e escuta toda a comunicação. Nesta Figura, a circunferência contínua representa o alcance da transmissão do nó Emissor, enquanto que a circunferência tracejada representa o alcance do nó destino.

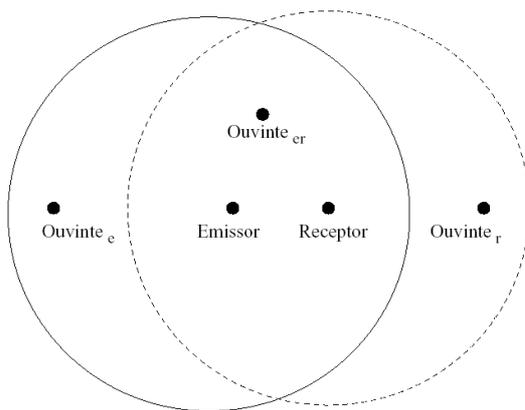


Fig. 6. Nós ouvintes.

A Tabela IV mostra a permanência dos três tipos de ouvintes nos estados de operação. Devido ao seu papel passivo na comunicação, os ouvintes em momento algum entram no estado Tx.

TABELA IV
PARCELA DE TEMPO QUE OS OUVINTES PASSAM EM CADA ESTADO.

Tamanho do Pacote (bytes)	ouvinte _{er}		ouvinte _e		ouvinte _r	
	t _{oc}	t _{Rx}	t _{oc}	t _{Rx}	t _{oc}	t _{Rx}
160	0,23	0,77	0,59	0,41	0,64	0,36
512	0,20	0,80	0,51	0,49	0,69	0,31
800	0,18	0,82	0,46	0,54	0,72	0,28
1000	0,17	0,83	0,43	0,57	0,74	0,26
1500	0,14	0,86	0,37	0,63	0,77	0,23

Deve ser feita uma ressalva no tocante aos nós ouvintes que recebem tráfego originado pelo emissor. Como estes nós não estão realmente interessados no conteúdo do quadro de dados, mas apenas nas

informações fornecidas pelos quadros RTS e CTS, eles podem economizar energia durante a transmissão do quadro de dados. Esta redução pode ser significativa, especialmente para interfaces enviando dados a taxas mais baixas. Os resultados de Feeney e Nilsson [9] mostram que a potência consumida neste período pode ser menor que P_{oc} para interfaces trabalhando a 2Mbps. No entanto, para as interfaces operando a 11Mbps, consideradas nesta análise, esta economia não se mostra significativa, já que os coeficientes que multiplicam o tamanho do pacote para a obtenção do consumo total no modelo linear empregado são similares para receptores e nós ouvintes. Por este motivo, será considerado que os ouvintes se alternam sempre entre os estados Rx e ocioso.

Desta forma, a Figura 7 mostra ao longo da transmissão de um quadro completo os estados das interfaces dos nós envolvidos numa comunicação direta.

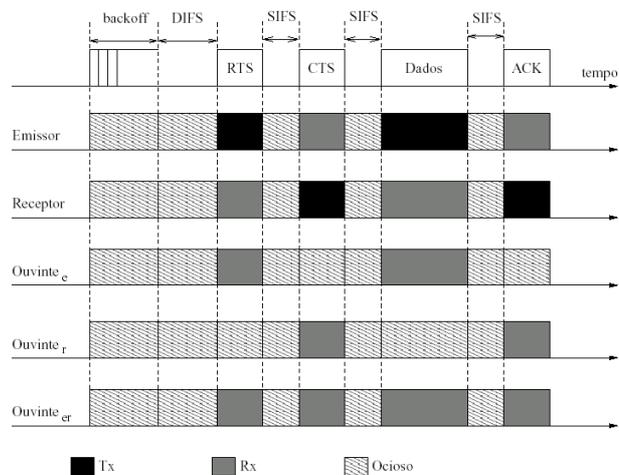


Fig. 7. Estados das interfaces dos nós numa comunicação direta.

III.2. COMUNICAÇÃO ATRAVÉS DE UMA CADEIA DE NÓS

Numa rede ad hoc a solução que os nós encontram para se comunicar com outros nós fora de seu alcance de transmissão é a formação de uma cadeia de encaminhamento cujas extremidades estão ligadas aos nós fonte e destino. Os pacotes são enviados salto-a-salto. Neste tipo de encaminhamento, os pacotes enviados consecutivamente competem entre si pelo meio, aumentando a contenção. Li *et al.* [14] mostram que a utilização ideal de uma cadeia de encaminhamento ad hoc é de 1/4 da taxa máxima a um salto. Para isto eles consideram um modelo de propagação em que se um pacote pode ser corretamente recebido numa distância *r*, esta transmissão seria capaz de causar interferência em transmissões distantes aproximadamente 2*r*.

Considerando a participação de um nó numa cadeia de encaminhamento ideal, este nó passaria 1/4 do tempo no papel de emissor e 1/4 do tempo no papel de receptor mostrados na Tabela II. No restante do tempo, ele seria ouvinte da comunicação ao redor dele na cadeia. Para determinar qual o tipo de ouvinte (Tabela IV) este nó será no restante do tempo, é necessário analisar os efeitos do raio de interferência. Se for considerado que o nível de sinal na região com distância *d* tal que *r* < *d* < 2*r* é suficiente para interferir numa conexão, mas não para

mudar sozinho o estado do nó ouvinte de ocioso para Rx, este nó será ¼ do tempo um ouvinte_r e ¼ do tempo um ouvinte_e. Neste Caso I, sua potência média consumida será:

$$P_m = \frac{1}{4}P_e + \frac{1}{4}P_r + \frac{1}{4}P_{or} + \frac{1}{4}P_{oe} \quad (6)$$

onde P_e, P_r, P_{or} e P_{oe} são as potências médias consumidas por nós dos tipos emissor, receptor, ouvinte_r e ouvinte_e respectivamente.

O Caso II considera que na distância *d* mencionada anteriormente o sinal de interferência é capaz de levar o nó ouvinte ao estado Rx de modo a tentar receber alguma informação (este é o modelo utilizado no simulador ns-2). Neste caso, sua potência média pode ser aproximada por:

$$P_m = \frac{1}{4}P_e + \frac{1}{4}P_r + \frac{1}{2}P_{oer} \quad (7)$$

onde P_{oer} é a potência média consumida por um nó ouvinte_{er}.

Os tempos de permanência em cada estado para os nós participantes de uma cadeia podem ser vistos na Tabela V. Como pode ser visto, o efeito da utilização de uma ou outra aproximação é apenas nas parcelas de tempo nas quais o nó permanece nos estados Rx e ocioso. A parcela de tempo no estado Tx, que apresenta o maior consumo, permanece inalterada.

TABELA V

PARCELA DE TEMPO QUE OS NÓS DE UMA CADEIA PASSAM EM CADA ESTADO.

Tamanho do Pacote (bytes)	Caso I			Caso II		
	t _{oc}	t _{Tx}	t _{Rx}	t _{oc}	t _{Tx}	t _{Rx}
160	0,42	0,19	0,39	0,23	0,19	0,58
512	0,40	0,20	0,40	0,20	0,20	0,60
800	0,39	0,21	0,40	0,18	0,21	0,61
1000	0,38	0,21	0,41	0,17	0,21	0,62
1500	0,36	0,22	0,42	0,14	0,22	0,64

IV. ANÁLISE DOS MÉTODOS DE ECONOMIA DE ENERGIA

Nesta seção serão analisados os ganhos potenciais de algumas técnicas de economia e balanceamento de energia em redes ad hoc. As técnicas analisadas são: o controle de potência de transmissão, o balanceamento de fluxos e a mudança do nó para o estado de dormência. Como base para análise serão utilizados os valores apresentados em [9] para o consumo de uma interface IEEE 802.11b operando a 11Mbps. Uma aproximação destes valores pode ser vista na Tabela VI.

TABELA VI

CONSUMO DE UMA INTERFACE IEEE 802.11B.

Estado	Potência (W)
Dormência	0,050
Ocioso	0,740
Rx	0,900
Tx	1,350

Para facilitar a comparação com o tempo de vida definido pela Equação 3, podemos expressar a potência consumida nos outros estados em função de P_{oc}. Desta

forma teremos de forma aproximada: P_{Tx} = 1,8 P_{oc}, P_{Rx} = 1,2 P_{oc} e P_d = 0,07 P_{oc}. Com base nestes valores, pode-se calcular P_m para as diversas situações mostradas anteriormente. A Figura 8 mostra a potência média consumida por nós (Equações 1 e 7 e Tabelas II e V) em comunicação direta e participando de cadeias (Caso II) em função de P_{oc}.

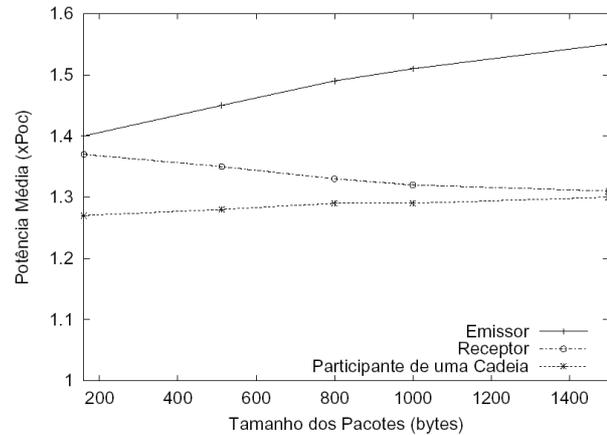


Fig. 8. Consumo de nós em função de P_{oc}.

Como pode ser visto na Figura 8, os consumos médios de Emissores e Receptores são muito próximos quando utilizados pacotes de 160 bytes. Isto ocorre porque a soma dos tempos de transmissão dos quadros de dados (t_{dados}) e RTS (t_{RTS}) é próxima da soma dos tempos de transmissão dos quadros CTS (t_{CTS}) e ACK (t_{ACK}). À medida em que o tamanho do pacote aumenta, a diferença entre os consumos dos nós Emissores e Receptores se diferenciam bastante devido ao aumento do tempo de transmissão do quadro de dados (t_{dados}).

Além disto, nós participando de cadeias de encaminhamento (considerando o Caso II) apresentam um tempo de vida entre 0,79 T_{oc} e 0,77 T_{oc} para os tamanhos de pacotes utilizados nesta análise. Isto resulta num total de dados úteis transmitidos (normalizando T_{oc} = 1) que varia entre 0,15Mb e 0,86Mb. A Figura 9 mostra o total normalizado de dados transmitidos utilizando diferentes tamanhos de pacotes para os cenários de comunicação direta e encaminhamento de pacotes através de uma cadeia de nós.

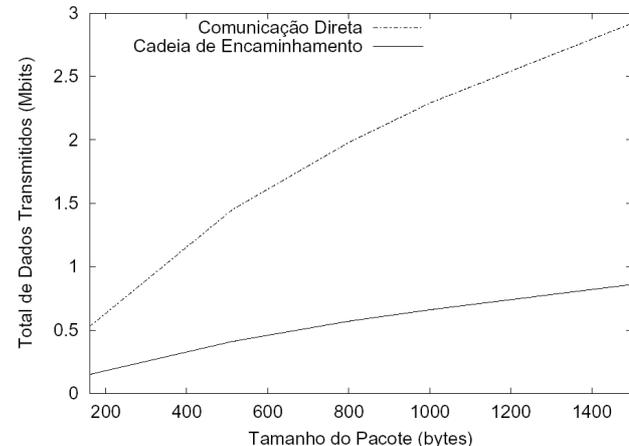


Fig. 9. Total normalizado de dados transmitidos.

Apesar do tempo de vida de nós participando de cadeias de encaminhamento ser maior que o tempo de

vida de nós em comunicação direta, a comunicação direta permite a transmissão de, pelo menos, 3,3 vezes mais dados que a comunicação através de uma cadeia para a mesma energia inicial.

Através de análises similares pode-se perceber, ainda, que nós na condição de ouvinte, possuem um tempo de vida entre 13%, quando utilizados pacotes de 160 bytes, e 15%, utilizando pacotes de 1500 bytes, menor que nós ociosos.

IV.1. CONTROLE DE POTÊNCIA DE TRANSMISSÃO

Alguns trabalhos buscam economizar energia através da redução da potência de transmissão utilizada [7] [10]. É claro que o raio de alcance da comunicação também diminui, sendo em alguns casos necessária a inclusão de mais saltos. Min e Chandrakasan analisam sob quais condições é interessante a utilização de múltiplos saltos a partir da redução da potência de transmissão [15]. É demonstrado que a utilização de dois saltos ao invés de um é vantajoso se a parcela que pode ser reduzida for maior que o custo fixo associado à comunicação. Nas interfaces sem fio, a parcela que pode ser reduzida é referente ao consumo do amplificador de potência RF. Esta parcela pode ser aproximada pela diferença entre P_{Tx} e P_{Rx} ou $0,6 P_{oc}$, com os valores utilizados na presente análise. Assim, considerando que a interface estará sempre no estado ocioso na melhor das hipóteses, o custo fixo pode ser estimado pela diferença entre P_{Rx} e P_{oc} que é $0,2 P_{oc}$. Nestas condições e desconsiderando a condição de ouvinte, para que seja vantajosa a utilização de dois saltos, a nova P_{Tx} deve ser menor que $1,3 P_{oc}$.

Quando pensamos na comunicação direta, é especialmente interessante a redução de P_{Tx} para o menor nível que permita a correta recepção da transmissão, já que todo o ganho é neste caso líquido. Nesta condição, o ganho limite para a potência consumida (considerando $lim(P_{Tx})= P_{Rx}$ - o que é um caso otimista já que considera que toda a diferença entre P_{Tx} e P_{Rx} é devido ao amplificador de RF) seria de $0,25 P_{oc}$ para emissores utilizando pacotes de 160 bytes e de $0,41 P_{oc}$ para emissores utilizando pacotes de 1500 bytes.

IV.2. TRANSIÇÃO PARA O ESTADO DE DORMÊNCIA

Esta técnica, embora muito usada em sistemas com algum tipo de administração centralizada, possui viabilidade limitada em redes ad hoc. Sua complexidade é maior que a das técnicas anteriores, já que necessita de algum mecanismo de controle para a entrada e saída dos nós do estado de dormência [16]. Além disso, um nó no estado de dormência deve confiar que seus vizinhos armazenarão pacotes endereçados a ele que venham eventualmente a ser enviados neste período.

Devido à grande diferença entre a potência consumida no estado ocioso e a potência consumida no estado de dormência, esta técnica apresenta o maior potencial de economia de energia. Sua melhor aplicação é em redes que ficam grandes períodos de tempo no estado ocioso. Nestes casos, o custo adicional do controle do mecanismo pode ser absorvido pelos ganhos.

Os pontos principais na utilização desta técnica são o mecanismo de controle, que não deve ser mais custoso do que a economia potencial, e a manutenção da capacidade da rede, que não deve ficar desconexa nem apresentar gargalos na nova topologia.

IV.3. ROTEAMENTO COM ENERGIA COMO MÉTRICA

Esta técnica [5] [3] [6] não tem, de modo geral, o objetivo de reduzir o consumo de energia total, mas sim de distribuir este consumo pela rede, resguardando quando possível nós com recursos mais escassos. Como o emissor e o receptor de uma comunicação não podem ser mudados, estes nós não se beneficiam desta técnica. Além disso, em comunicações diretas não há opção de caminho a ser utilizado, fazendo com que esta técnica não possa ser utilizada. Esta técnica pode beneficiar nós que participam de cadeias de encaminhamento.

A análise apresentada nesta seção considera inicialmente casos em que o tráfego é balanceado igualmente por n caminhos diferentes, fazendo com que os nós de cada caminho participem do encaminhamento de pacotes $\frac{1}{n}$ do tempo. Esta análise pode, no entanto, ser generalizada para situações em que os nós participem desigualmente do encaminhamento de pacotes. Neste caso, a fração $\frac{1}{n}$ utilizada para o cálculo da potência média balanceada do nó (P_{mbal}) passa a representar a parcela do tempo total que um nó específico participa de cadeias de encaminhamento.

Se considerarmos a existência de n caminhos separados (caminhos em que os nós de um caminho não sejam ouvintes do tráfego de algum dos outros caminhos), a potência média consumida pelo nó será:

$$P_{mbal} = \frac{(n-1)P_{oc}}{n} + \frac{P_m}{n} \tag{8}$$

onde P_m é a potência média consumida pelo nó no cenário considerado sem a utilização do balanceamento e P_{mbal} é a potência média consumida pelo nó no cenário considerado com a utilização do balanceamento.

No entanto, como mostra a Figura 10, não existem caminhos totalmente separados, já que os nós das extremidades da cadeia de encaminhamento devem estar invariavelmente no alcance da fonte ou do destino da comunicação. Nesta Figura, os nós E_{f1} e E_{d1} são os nós das extremidades do caminho 1 ligados ao nó fonte e ao nó destino respectivamente. O mesmo ocorre com os nós E_{f2} e E_{d2} em relação ao caminho 2. As circunferências ao redor do emissor e do receptor representam seus raios de alcance.

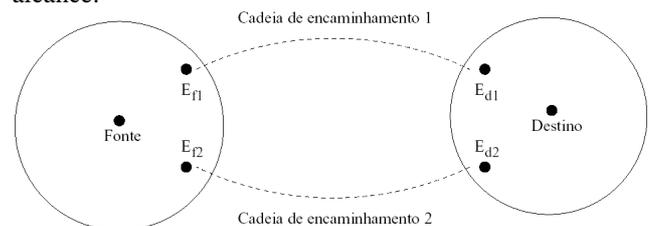


Fig. 10. Nós das extremidades das cadeias de encaminhamento.

Para estes nós, a potência média consumida quando não fizerem parte da cadeia de encaminhamento ativa não será P_{oc} . Eles serão ouvintes das transmissões que ocorram perto das extremidades da cadeia. Assim, a potência média consumida por estes nós (P_{ex}) será

$$P_{ex} = \frac{1}{4}P_{oc} + \frac{1}{4}P_{oe} + \frac{1}{2}P_{oer} \quad (9)$$

e sua potência média consumida resultante do balanceamento ($P_{mbal/ex}$) será:

$$P_{mbal/ex} = \frac{(n-1)P_{ex}}{n} + \frac{P_m}{n} \quad (10)$$

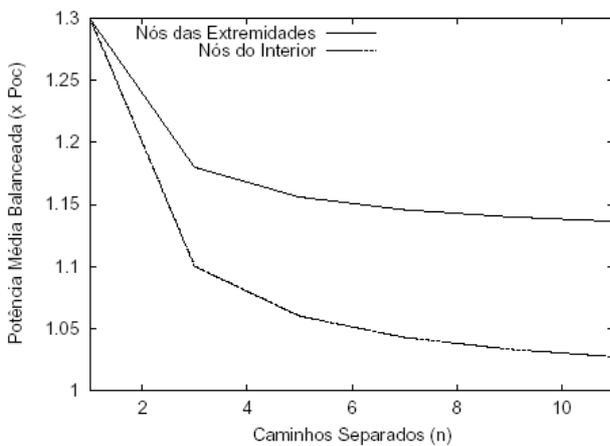
Numa cadeia de nós transmitindo pacotes de 1500 bytes, por exemplo, considerando P_m para o Caso II, P_m seria $1,3 P_{oc}$ e P_{ex} seria $1,12 P_{oc}$. Nestas condições, o limite para o ganho em tempo de vida (Equação 2) obtido em função do número de cadeias de encaminhamento separadas ($n \rightarrow \infty$) é 30% para os nós no interior da

cadeia e 16% para os nós nas extremidades. A evolução das potências médias balanceadas em função do número de caminhos separados nestas condições pode ser vista na Figura 11.

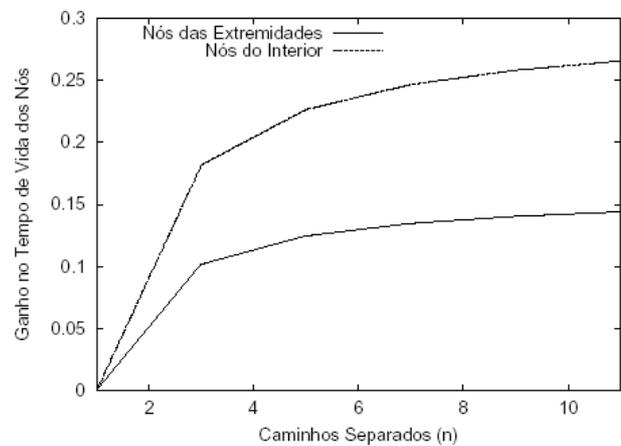
Se for considerada a utilização de caminhos disjuntos, porém não separados, os nós integrantes das cadeias podem ser considerados como nós do tipo ouvinte_{er} nos períodos em que sua cadeia estiver inativa. Desta forma, sua potência mínima consumida será igual a P_{oer} . A potência média resultante do balanceamento ($P_{mbal/disj}$) será:

$$P_{mbal/disj} = \frac{(n-1)P_{oer}}{n} + \frac{P_m}{n} \quad (11)$$

Se o tamanho dos pacotes transmitidos for de 1500 bytes como na análise anterior, o limite ($n \rightarrow \infty$) para o ganho em tempo de vida (Equação 2) dos nós é de 11%. A variação deste ganho em função do número de caminhos disjuntos pode ser vista na Figura 12.



(a) Potência Média Consumida



(b) Ganho no Tempo de Vida

Fig. 11. Efeitos do número de caminhos separados

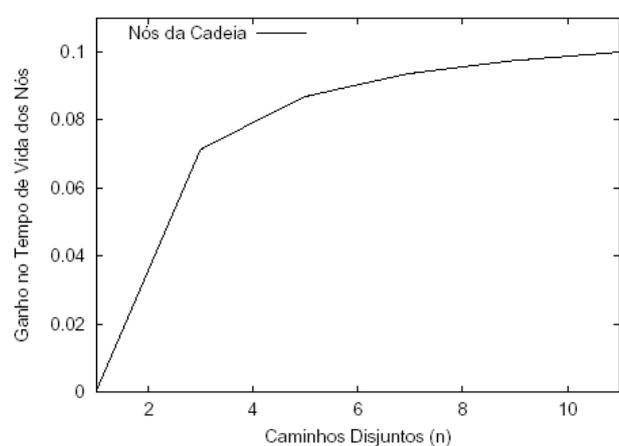
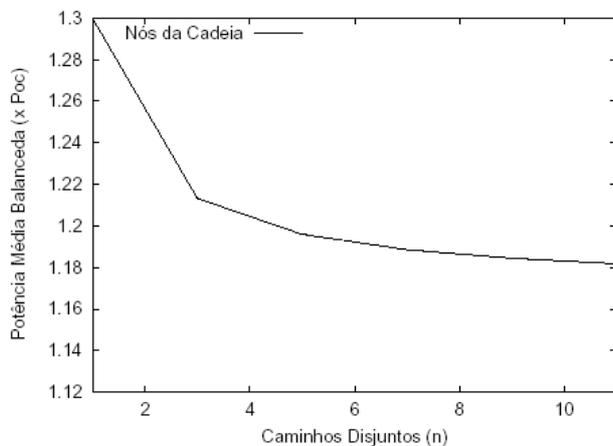


Fig.12. Efeitos do número de caminhos disjuntos

V. CONCLUSÕES

Este artigo analisou o consumo de energia dos nós integrantes de uma rede ad hoc levando em consideração as características do método de acesso e as interações entre os nós, necessárias ao encaminhamento de pacotes. Levando estes dois fatores em consideração, foi calculada a parcela do tempo durante a qual as interfaces de rede dos nós permanecem nos estados Tx, Rx e ocioso de modo a possibilitar o cálculo da potência média consumida. A potência média consumida permite estimar o tempo de vida de um nó.

Os resultados mostram que ao se utilizar pacotes maiores, as interfaces passam mais tempo no estado Tx, aumentando a potência média consumida e, conseqüentemente, reduzindo o tempo de vida do nó. No entanto, a redução do tempo de vida é muito menor do que o ganho obtido em vazão devido a utilização de pacotes maiores. Desta forma, pode-se concluir que uma maior eficiência em bits por joule é obtida com a utilização de pacotes maiores.

A análise também demonstra que um nó ouvinte, ou seja, um nó que esteja dentro do raio de cobertura de uma comunicação entre outros dois nós, tem o tempo de vida reduzido em até 15%.

São analisadas, ainda, técnicas de economia e balanceamento do consumo de energia. Para os modelos utilizados é mostrado que, em certas circunstâncias, o uso de múltiplos saltos e a redução da potência de transmissão podem ser vantajosos. Mostra-se, ainda, que o limite superior do ganho obtido para o tempo de vida dos nós com o balanceamento de fluxos em redes ad hoc é de 30% para nós no interior de cadeias de encaminhamento e de 16% para nós nas extremidades das cadeias. Por outro lado, se forem considerados caminhos disjuntos com alcances de rádio sobrepostos, o ganho máximo é limitado a 11%.

REFERÊNCIAS

- [1] S. Singh, M. Woo e C. S. Raghavendra, "Power-aware routing in mobile ad hoc networks", em *ACM MOBICOM*, pp. 181–190, outubro 1998.
- [2] J.-H. Chang e L. Tassiulas, "Energy conserving routing in wireless ad-hoc networks", em *IEEE INFOCOM*, pp. 22–31, março 2000.
- [3] W. Yu e J. Lee, "DSR-based Energy-aware routing protocols in ad-hoc networks". ICWN, junho 2002.
- [4] D. A. Maltz, D. B. Johnson, Y. C. Hu. e J. G. Jetcheva, "The dynamic source routing protocol for mobile ad hoc networks", *Internet Draft, draft-ietf-manet-dsr-09.txt*, abril 2003.
- [5] D. de O. Cunha, L. H. M. K. Costa e O. C. M. B. Duarte, "Um mecanismo de roteamento para o consumo balanceado de energia em redes móveis ad hoc", em *XXI Simpósio Brasileiro de Redes de Computadores*, pp. 747–760, maio 2003.
- [6] W. Cho e S.-L. Kim, "A fully distributed routing algorithm for maximizing lifetime of a wireless ad hoc network", em *IEEE Conference on Mobile and Wireless Communications Networks*, 2002.
- [7] P.-J. Wan, G. Calinescu, X. Li e O. Frieder, "Minimum-energy broadcast routing in static ad hoc wireless networks", em *IEEE INFOCOM*, pp. 1162–1171, março 2001.
- [8] N. K. Singh, R. Simha e B. Narahari, "Energy balance in wireless networks using connection segmentation and range control", em *IEEE WCNC*, 2003.
- [9] L. M. Feeney e M. Nilsson, "Investigating the energy consumption of a wireless network interface in an ad hoc networking environment", em *IEEE INFOCOM*, março 2001.
- [10] J. Monks, J.-P. Ebert, W.-M. W. Hwu e A. Wolisz, "Energy saving and capacity improvement potential of power control in multi-hop wireless networks", em *Computer Networks*, vol. 3, no. 41, pp. 313-330, fevereiro 2003.
- [11] IEEE, "Wireless LAN medium access control(MAC) and physical layer (PHY) specifications". IEEE Standard 802.11, 1999.
- [12] IEEE, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications: Higher-speed physical layer extension in the 2.4 GHz band". IEEE Standard 802.11b, 1999.
- [13] K. Fall e K. Varadhan, em *The ns Manual*. UC Berkeley, LBL, USC/ISI, and Xerox PARC, abril 2002. Available at <http://www.isi.edu/nsnam/ns/ns-documentation.html>.
- [14] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee e R. Morris, "Capacity of ad hoc wireless networks", em *Mobile Computing and Networking*, pp. 61-69, 2001.
- [15] R. Min e A. Chandrakasan, "Top Five Myths about the Energy Consumption of Wireless Communication". *ACM MOBICOM*, setembro 2002. Poster.
- [16] B. Chen, K. Jamieson, H. Balakrishnan e R. Morris, "Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks", em *ACM MOBICOM*, julho 2001.

Adaptive Bit and Power Allocation for Rate and Margin Maximization in V-BLAST System

¹Dror Meiri and ²Irving Kalet

¹ Aelis Photonics, Netanya, Israel (email: dmeiri@netvision.net.il)

² Faculty of Electrical Engineering, Technion – Israel Institute of Technology, Haifa, Israel (email: kalet@actcom.co.il)

Abstract— This paper describes the "Water-Pouring" solution for wide-band multitone version of the Vertical Bell Labs Layered Space-Time (V-BLAST) architecture. It is shown that the multitone concept may be combined with the Multiple-Input-Multiple-Output (MIMO) V-BLAST concept for use on a frequency-selective channel. We present an optimal power and bit loading algorithm which attempts to optimize the system performance by enhancing either the data rate or the QoS (error rate), while keeping the other fixed. In both methods, a power constraint exists. Simulation results of the proposed system and algorithm show that adaptive loading enables efficient and reliable communication even with low SNRs. It also shows significant improvement in both methods as compared to a basic OFDM V-BLAST system, which has no a-priori knowledge of the channel characteristics. For example, a system with four transmitters and receivers that uses a 256 multitone symbol doubles the nominal bit rate at SNR of 33dB (for maximum bit rate) and achieves a gain of 19dB at BER of 2×10^{-4} (for maximum QoS).

Index Terms—fading channels, multi-input-multi-output (MIMO) systems, orthogonal frequency division multiplexing (OFDM), wireless communications.

I. INTRODUCTION

The demand for high data rate communication systems, and the lack of available bandwidth has led to much research effort into MIMO systems using multiple transmit and receive antennas. It was shown in [1] that MIMO systems can improve the capacity of rich-scattering Rayleigh fading channels, by a factor which is linearly proportional to the minimum of the number of the transmit and receive antennas. The V-BLAST [2] architecture has been proposed for realizing high spectral efficiencies over flat Rayleigh fading indoor wireless channels. Turbo coding in V-BLAST systems was proposed in [17].

Multitone transmission [3,4], or OFDM, has been used in a number of wireline and wireless systems. OFDM eliminates the need of a complex equalizer (in wireless systems) for overcoming ISI caused by delay spread or frequency-selective fading. However, in [3,4] it was shown that for a single-input-single-output (SISO) system which operates over a frequency-selective channel, if the transmitter "knows" the channel characteristics, the "water-pouring" multitone (MT) technique can be used to

optimize the transmitted bit rate and approach the channel capacity. Adaptive bit and power allocation algorithms [11] were shown to achieve efficient use of the "water-pouring" solution in wireline ADSL and VDSL systems.

In [1,12,13,18], the channel capacity for the MIMO system has been analyzed and discussed. Raleigh and Cioffi [7] have developed a coding structure that achieves the capacity and a technique that approaches this structure.

Few works have considered the usage of V-BLAST in MIMO-OFDM systems. Adaptive modulation for such systems has been presented by Ng et al [14]. This work assessed the maximum margin concept only. Meiri and Kalet [8,15,16] have developed an adaptive loading algorithm for the Wide-Band Multitone version of V-BLAST for both maximum rate and maximum margin criteria.

In this paper we present adaptive loading algorithms that optimize either the data rate or the margin (known also as Quality of Service (QoS) maximization), while keeping the other fixed. In both methods, a power constraint exists. Both optimization problems result in (different) power and bit allocations, i.e. each sub-channel carries a different amount of data and transmitted with different power. For the maximum margin approach, we show that for an equal number of antennas on both sides of the link, this algorithm results in improvement of several dBs for BERs of 10^{-3} and 10^{-4} compared to a MIMO system with the best performance that has no channel knowledge at the transmitter, irrespective of the number of antennas. For the maximum bit rate, we investigate two types of error constraints: (a) fixed error rate for all SNRs and (b) error rate figures of a typical OFDM V-BLAST system having the same system parameters (and that does not use adaptive loading). The first constraint results in a practical system that implements the Multi-Tone-Multi-Antenna (MTMA) technique [8]. The second one maximizes the communication efficiency of any given system. We show that for an equal number of antennas on both sides of the link, and a fixed error rate constraint, using the adaptive loading algorithm enables efficient and reliable communication in low SNRs, compared to MIMO systems with the best performance which has no channel knowledge at the transmitter. For the case which there is a nominal error rate constraint, we show that at high SNR values the system more than doubles the bit rate.

The paper is organized as follows: Section 2 describes the system and channel model, Section 3 describes the V-BLAST system, Section 4 presents the wideband

multitone version of V-BLAST including the adaptive bit and power allocation algorithm. Section 5 gives the simulation results and Section 6 presents a summary and conclusions.

II. SYSTEM AND CHANNEL MODEL

A single data stream is split into multiple sub-streams that are transmitted simultaneously from an array of collocated M antennas. All the sub-streams occupy the same (and the entire) bandwidth, and have the same symbol duration. The sum of the powers of all the transmitters is constrained. There are N ($N \geq M$) receiving antennas. The complex term $h_{ij}(t)$ represents the independent channel impulse response (with a Rayleigh distributed amplitude) for the signal, $s_i(t)$, transmitted from the i th antenna, and received by the j th receiver antenna. Accordingly, the complex valued signal, $r_j(t)$, received at receiver j is given by

$$r_j(t) = \sum_{i=1}^M h_{ij}(t) * s_i(t) + n_j(t) \quad (1)$$

where $*$ stands for convolution, $i \in 1..M$, $j \in 1..N$ and $n_j(t)$ is a zero-mean complex AWGN signal, with PSD of $N_0/2$ watts/Hz. The channel frequency response may be flat, or frequency-selective (if the channel suffers from ISI [5]). It is assumed that the receiver has perfect channel knowledge. This may be the case for a slowly-changing mobile channel.

III. V-BLAST OVERVIEW

The MIMO V-BLAST system [2] has been used in frequency-nonselctive channels. The received signals are sequentially decoded using three steps: (a) Nulling the interference from the yet undetected symbols, (b) Symbol detection according to the optimal detection rule, (c) Cancellation of the previously detected symbols. The order of detection is significant and there is an optimal ordering rule.

The Zero-Forcing V-BLAST algorithm [2] is described below:

a. Initialization

$$i = 1 \quad (2a)$$

$$G_1 = \text{pinv}(H) \quad (2b)$$

$$K_1 = \arg \min_j \|(G_1)_j\|^2 \quad (2c)$$

b. Recursion

$$w_{K_i} = (G_i)_{K_i} \quad (2d)$$

$$y_{K_i} = w_{K_i}^T r_i \quad (2e)$$

$$\hat{s}_{K_i} = \text{slice}(y_{K_i}) \quad (2f)$$

$$r_{i+1} = r_i - \hat{s}_{K_i}(H)_{K_i} \quad (2g)$$

$$\begin{aligned} H_{K_1} &= H_{K_2} \dots \\ &= H_{K_i} = \text{zeros}(1, N) \end{aligned} \quad (2h)$$

$$G_{i+1} = \text{pinv}(H) \quad (2i)$$

$$K_{i+1} = \arg \min_{j \in \{K_1 \dots K_i\}} \|(G_{i+1})_j\|^2 \quad (2j)$$

$$i = i + 1 \quad (2k)$$

where H is the channel matrix of size $M \times N$, w is the nulling operator, r the received vector, y the receiver's estimate, i the index of the transmitted signal that is subject to the current detection, and K_i is a permutation of i and refers to the optimal detection ordering. Slice operation implements the maximum likelihood (ML) symbol decoding, $\text{zeros}(1, N)$ is a vector of length N whose elements have zero value, and pinv is the pseudo inverse operator.

IV. A WIDEBAND MULTITONE VERSION OF V-BLAST

In the Multi-Carrier Modulation (MCM) technique [3,4] a wideband signal is composed of many narrowband QAM signals ("tones" or "frequency-bins"). Hence, each individual tone sees an essentially frequency-nonselctive channel, and a simple one-tap equalizer can nullify the channel frequency distortion. MCM can also be used in the system defined in Section Two, and accordingly the many narrowband received MIMO signals can be decoded using V-BLAST. A base-band diagram of this system is shown in Figure 1.

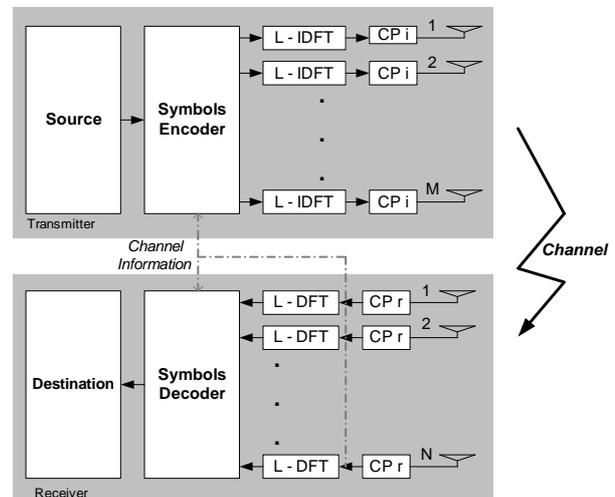


Fig. 1. Block diagram of multitone version of V-BLAST.

This is an extension of the Discrete Multitone (DMT) implementation [11] of the multitone system used in ADSL and VDSL. There are L sub-channels in each L-DFT and L-IDFT. The CP blocks at the transmitter and the receiver represent the introduction and removal of the cyclic prefix, respectively.

If the transmitter has no channel knowledge, the MCM technique becomes a simple OFDM system in which the same modulation is used at each transmitting antenna in each frequency. The total error rate is the same as that of V-BLAST over a frequency-nonselctive channel.

Knowledge of the channel characteristics at the transmitter(s) enables the transmitter(s) to adapt the transmitted signal to the channel characteristics in such a way that the system performance will be improved. The improvement can be manifested by enhancing either the data rate or the QoS (error rate) while keeping the other constant. These are the solutions of the following problems:

- The maximum bit rate optimization problem:

$$R = \max_{b_{m,l} \in D} \sum_{l=1}^L \sum_{m=1}^M b_{m,l}$$

under the total power constraint:

$$\sum_{l=1}^L \sum_{m=1}^M P_{m,l} \leq P_{Total}$$

- The maximum QoS optimization problem:

$$P_{Total} = \min_{b_{m,l} \in D} \sum_{l=1}^L \sum_{m=1}^M P_{m,l}$$

under the constant bit rate constraint:

$$R = \sum_{l=1}^L \sum_{m=1}^M b_{m,l}$$

where $b_{m,l}$ and $P_{m,l}$ are the number of bits and the power allocated for the m -th antennas at the l -th frequency-bin, respectively. D is the set of all possible values of $b_{m,l}$. It is important to note that even though the last problem is formulated to minimize the overall transmit power for a given QoS requirement, the same solution can be applied to maximize the QoS for a given overall transmit power. This can simply be achieved by increasing the power proportionally for all sub-carriers, while using the same set of $\{b_{m,l}\}_{m=1,l=1}^{M,L}$ [6].

In the following, we present an optimal adaptive bit and power allocation algorithm. It is a practical way to implement the MTMA technique [8] and to optimize the system performance under the two criterias given above.

Let $f(b)$ denote the single sub-channel required received power (in energy per symbol time) needed for reliable reception of b information bits per symbol with a unity channel gain. For a square QAM constellation, $f(b)$, is given by [5]:

$$f(b) \cong \left\{ Q^{-1} \left(\frac{P_r(e)}{K} \right) \right\}^2 \frac{(2^b - 1)N_0}{3} \quad (3)$$

where $p_r(e)$ is the symbol error probability, $2 \leq K \leq 4$ and Q^{-1} is the inverse of the Q-function.

De-Souza [9] proved that a given allocation is an optimal solution to the problems above if and only if any change in the allocation does not require less transmitted energy ("Efficiency") and all resources (power for the maximum bit rate, and bit rate for the maximum QoS criterions) are exploited ("Tightness").

For QAM, $f(b)$ is a convex, increasing function of b with $f(0)=0$. Hence, the loading algorithm that "puts every increment of the transmitted power where it is most effective" results in an efficient allocation [9].

In V-BLAST each symbol is decoded separately: The symbols that have previously been decoded are already cancelled, and the symbols that have not been detected yet are seen as interferers and are nulled out. This leaves us with a single symbol, which is ML decoded. The other symbols (out of M symbols) are also decoded, in their turn. The conclusion from the above is that the convexity of $f(b)$ holds also in the spatial dimension imposed by V-BLAST, and hence an optimal allocation can be reached as before.

In order to maintain the required error rate or QoS at the receiver, the transmit power allocated to the l -th sub-carrier of the m -th transmitting antenna must be equal to

$$P_{m,l} = G_{m,l}(H) f(b_{m,l}) \quad (4)$$

where G is a function of the channel and implements the reciprocal of the corresponding channel energy gain. Recalling the V-BLAST, G can be given by:

$$G_{m,l}(H) = \|w_{m,l}\|^2 \quad (5)$$

where $w_{m,l}$ for an arbitrary subcarrier index l is given in (2d). Note that for SISO system $w_{m,l}$ becomes $1/|H_l|^2$.

As the power needed to transmit a certain number of bits in a single sub-channel (out of the $M \times L$ sub-channels) is independent of the number of bits allocated to other sub-channels, assigning one bit one at a time to the sub-carrier that requires the least additional power is "efficient". The bit allocation process will be completed when all power is allocated, for the maximum bit rate approach, or when all R bits are assigned for the maximum QoS approach and hence the allocation is "tight". The suggested algorithm is given by:

Step a – Initialization

For every sub-carrier m , and every transmitter l

$$b_{m,l} = 0, P_{m,l} = 0 \quad (6a)$$

Step b – Bit and Power Assignment Iterations

While $\sum_{m=1}^M \sum_{l=1}^L P_{m,l} \leq P_{Total}$ (for maximum bit rate)

or

While $\sum_{m=1}^M \sum_{l=1}^L b_{m,l} \neq R$ (for maximum QoS)

For every sub-carrier l and every transmitter antenna m ,

$$\Delta P_{m,l}(b_{m,l} + 1) = \left\| w_{m,l} \right\|^2 * [f(b_{m,l} + 1) - f(b_{m,l})] \quad (6b)$$

$$\{\hat{m}, \hat{l}\} = \arg \min_{m,l} \Delta P_{m,l} \quad (6c)$$

$$b_{\hat{m}, \hat{l}} = b_{\hat{m}, \hat{l}} + 1 \quad (6d)$$

$$P_{\hat{m}, \hat{l}} = P_{\hat{m}, \hat{l}} + \Delta P_{\hat{m}, \hat{l}} \quad (6e)$$

Step c – Finish

$b = \{b_{m,l}\}_{m=1,l=1}^{M,L}$ and $P = \{P_{m,l}\}_{m=1,l=1}^{M,L}$ are the bit and power allocation solution.

For the maximum QoS approach, after allocation is finished, the total power should be normalized to accommodate the power budget, as explained before. Also note that for the maximum QoS, the calculation in (6b), can be replaced with a more convenient expression:

$$\Delta \tilde{P}_{m,l}(b_{m,l} + 1) = \left\| w_{m,l} \right\|^2 * [2^{b_{m,l}+1} - 2^{b_{m,l}}] \quad (7)$$

where $w_{m,l}$ is the m-th row of the pseudo-inverse of the l-th sub-channel matrix.

As in the V-BLAST reception scheme, the order of the calculation of $w_{m,l}$ for a given sub-channel is of importance. In the V-BLAST receiver the component with the smallest post-detection signal to noise ratio (PDSNR) dominates the error performance of the system, and thus choosing the symbol with best PDSNR at each stage in the detection process leads to the globally optimal ordering [2]. Although the allocation algorithm aims to achieve the same PDSNR for all received sub-channels, optimal ordering at the allocation process should be done in a similar manner to that of the V-BLAST receiver because the integer constraints result in different PDSNR for two sub-channels allocated with the same number of bits. Summarizing, the algorithm for calculating the weight vector $w_{m,l}$ for the lth sub-carrier is a sub-section of V-BLAST algorithm, obtained by using equations (2a),(2b),(2c),(2d),(2h),(2i),(2j) and (2k).

One of the properties of the V-BLAST system is that it exploits the reception diversity in MIMO systems where $M \leq N$ to achieve high QoS performance. Note that the algorithm will inherently achieve results that are equal or better than any reception diversity system with $M \leq N$, since this solution is a subset of the solution space of the algorithm we propose.

The algorithm is optimal also for bit and power allocation in a MIMO system in which $M < N$. Note, however, that the calculation of $f(b)$ changes since the required energy for receiving b bits gets smaller because of the diversity enhancement. Naturally, as the ratio between N and M gets larger, the improvement achieved by the loading will get smaller, because of the channel averaging effect.

The algorithm is optimal also for V-BLAST systems that operate over frequency non-selective fading channel.

In this case there is a single carrier, accordingly $L=1$ in all equations. If we take only one transmitter and receiver ($M=1$), the algorithm becomes the well known "greedy algorithm" [9]

The loading algorithm described so far handles MIMO systems that operate over time invariant channels. Clearly, the same mechanism can be used over time-varying channels. But considering the complexity of the loading algorithm, such a system might be impractical. However, assuming that (a) the time variant channel can be divided into time slots in which the channel can be considered as constant or time invariant, and (b) that the channels corresponding to adjacent time slots have partial correlation, it will be useful taking advantage of the extended "Efficientize" algorithm given in [8] and to use the previous allocation as an initialization for the current channel. Accordingly the bits allocated in every sub-channel will be increased or decreased at every iteration, rather than increased only when the initial allocation is zero.

V. SIMULATION RESULTS

The following examples shows performance gain of three systems: The first is required to transmit a maximum number of bits, with the symbol error rate constrained to be smaller or equal to 2×10^{-3} (maximum bit rate criterion). The second system is required to transmit 4096 bits per symbol with the best error rate (maximum QoS criterion). The third system is required to transmit maximum number of bits, with the symbol error rate constrained to be smaller or equal to the symbol error rate of the OFDM V-BLAST system. All the three systems use a 256 multitone symbol, and there are four transmitting and four receiving antennas. The symbols are chosen from square QAM constellation and thus an even number of bits are Gray mapped onto the symbols. Each simulation run assumes that the channel parameters for that run are known at the receiver and at the transmitter (when relevant). These parameters are sampled from the Rayleigh distribution pertinent to the delay profile. The profile we use for these examples is the DVB-T [10] portable channel. The SNR figures at the abscissa, are relative, and are arbitrarily defined as follows: Each receiving antenna receives the superimposed signal from all M transmitters. This signal has a power of P_{rx} per received symbol. Accordingly the received SNR (at each antenna) is defined as $SNR = 10 \log_{10} 2P_{rx}/N_0$. The relation between the total transmitted energy and the received energy is given by $P_{rx} = P_{tx}/E(r)^2$, where r is a Rayleigh distributed random variable. P_{tx} is divided between the M transmitters over the L frequency-bins. The graphs below were obtained by averaging the results over many channel realizations.

Figure 2 refers to the first example (maximum rate at constant error rate constraint) and shows two curves: The first is for equal bit and power allocation and the fixed error rate constraint. The second shows results for our adaptive loading algorithm. In this case the number of bits per antenna per tone varies from zero to eight, and the energy per tone is also variable. The optimized results

show significant improvement, compared to the first case. As expected from the water-pouring solution properties, the smaller the SNR is, the greater the gain. As can be seen, at SNR below 15 dB the system with no loading collapses. Adaptive loading, however, enables communication in such conditions. Note also that for SNR of 30 dB the bit rate gain is approximately 1.75.

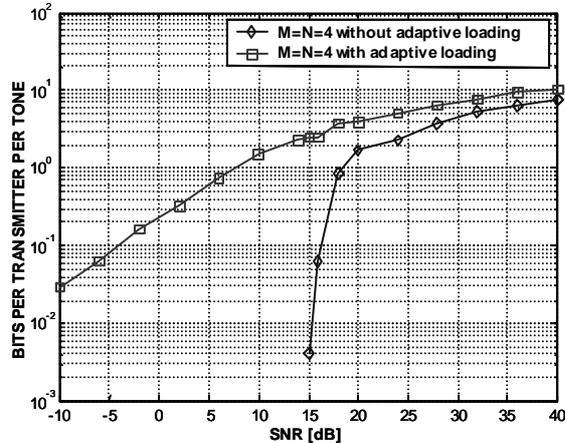


Fig. 2. Bits per transmitter per tone vs. SNR with and without adaptive loading, with 256 multitone symbol over a DVB-T channel, $M=N=4$, $P_r(e) = 2 \times 10^{-3}$.

Figure 3 refers to the second example (maximum QoS) and also shows two curves. The first is for a V-BLAST OFDM system with an equal bit and power allocation: four bits per antenna per tone. The second shows results for our adaptive loading algorithm. In this case the number of bits per antenna per tone varies from zero to eight, and the energy per tone is also variable. The optimized results show significant improvement, compared to the first case. For example there are gains of 13 and 19 dB, respectively, at BERs of 10^{-3} and 2×10^{-4} . Note also that the higher the SNR is, the greater the gain.

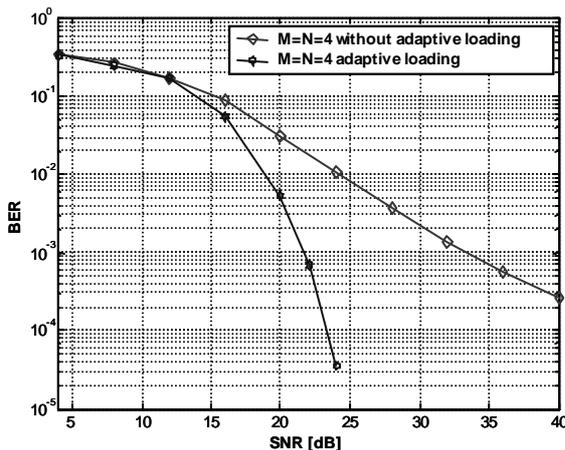


Fig. 3. BER vs. SNR with and without adaptive loading for 4096 bits per 256 multitone symbol (constant bit rate) at s MIMO system with $M=N=4$.

Figure 4 refers to the third example (maximum rate at OFDM V-BLAST error rate constraint) and we present the potential bit rate gain vs. SNR. The system is required to have at any SNR the same symbol error rate as that of a system that uses no adaptive loading (OFDM V-BLAST). For example at $BER=2 \times 10^{-3}$, the required SNR in system

without adaptive loading is 30 dB. This system's performance can be improved by our algorithm in two senses: (a) lowering the required SNR by approximately 10 dB (as demonstrated in Figure 3) or alternatively (b) enhancing the bit-rate by approximately 1.75 times the original rate (as also shown in Figure 4). Moreover, at SNR of 33dB one can have the same system performance as an equivalent system with 8 transmitters and 8 receivers that does not use adaptive loading. It is important to note that all the results of this section are achieved without coding.

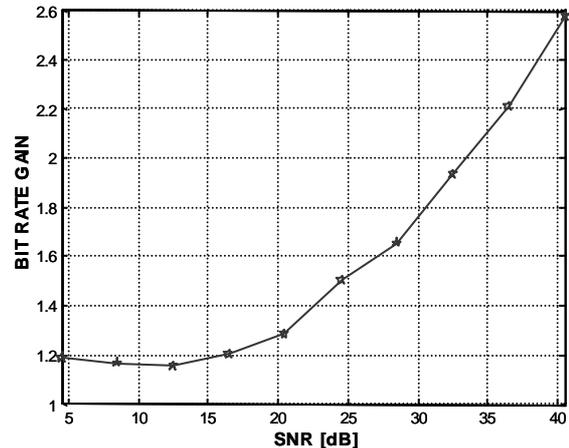


Fig. 4. Potential bit rate gain vs. SNR with adaptive loading compared to OFDM V-BLAST with 4096 bits per 256 multitone symbol over a DVB-T channel, $M=N=4$.

VI. SUMMARY AND CONCLUSIONS

We have presented an algorithm for adaptive bit and power allocation in V-BLAST systems operating in any slow fading channel. The loading algorithm is optimal for a maximum bit rate criterion and for maximum QoS (maximum margin) criterion. Simulations of this system show significant improvement for both criteria: By adopting the maximum rate approach it is possible either to achieve efficient and reliable communication with low SNRs (which is not achievable in the original V-BLAST system), or to increase the bit rate. By adopting the maximum QoS approach the power margin is improved with the SNR.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for the helpful comments and feedback.

REFERENCES

- [1] G. J. Foschini and M. J. Gans, "On limits of wireless communication in a fading environment when using multiple antennas", *Wireless Personal Comm.* (6), Kluwer Academic Publishers, Mar. 1998, pp. 311-335
- [2] P.W. Wolniansky, G.J. Foschini, G.D. Golden and R.A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel," *Proc. URSI ISSSE '98, Pisa, Italy, 1998*, pp. 295-300.

- [3] I. Kalet, "The multitone channel," IEEE Trans. Communication, vol.37, pp 119-124, Feb. 1989.
- [4] J.A.C. Bingham, "Multicarrier modulation for data transmission: An idea whose time has come," IEEE Communication Magazine, pp. 5-14, May 1990.
- [5] J. G. Proakis, Digital Communication. New York: McGraw-Hill, 2001.
- [6] C.Y. Wong et al, "Multiuser OFDM with adaptive subcarrier, bit and power allocation", IEEE J. Select. Areas Comm. Vol. 17, Oct. 1999, pp.1747-1758.
- [7] G.G. Raleigh, J.M. Cioffi, "Spatio-temporal coding for wireless communication", IEEE Trans. Communication, Vol.46, No.3, Mar. 1998, pp 357-366.
- [8] D. Meiri, "Wide Band Mutitone Version of V-BLAST", MSc Thesis, Department of Electrical Engineering, Technion, Haifa, Israel, September 2002.
- [9] J.C. De Souza, "Discrete Bit Loading For Multicarrier Modulation Systems", Phd Thesis, Department of Electrical Engineering, Stanford University, May 1999.
- [10] ETSI, "Digital Video Broadcasting (DVB); Framing structure, channel coding and modulation for digital terrestrial television (DVB-T)", ETS 300744, March 1997
- [11] P.S. Chow, et al. "A discrete multitone transceiver for HDSL applications", IEEE J. Select. Areas Comm. Vol. 9, Aug. 1991, pp.895-908.
- [12] E. Telatar, "Capacity of multi-antenna gaussian channels", Eur. Trans. Telecomm., Vol. 10, No. 6, November-December 1999, pp. 585-595
- [13] Chen-Nee Chuah, "Capacity of Multi-Antenna Array Systems", MSc. Thesis, University of California, Berkley CA, January 25, 2000
- [14] K.-W. Ng, et al. "Iterative bit and power allocation for V-Blast based OFDM MIMO system in frequency selective channels", WCNC '02, Mar. 2002, pp.271-275.
- [15] D. Meiri, I. Kalet "Wideband Multitone version of V-BLAST", 7th international workshop on OFDM, September 2002, Hamburg, Germany
- [16] D. Meiri, I. Kalet "Adaptive Loading for Multitone version of V-BLAST", 22nd ISRAEL IEEE Conference, December 2002, Tel-Aviv, Israel
- [17] M. Sellathurai and S. Haykin, "TURBO BLAST for high speed wireless communication", WCNC '00, September 2000.
- [18] H. Bolckei et al. "On the capacity of OFDM-based spatial multiplexing systems", IEEE Trans. Communication, vol.50, pp 225-234, Feb. 2002.



Dror Meiri (S'00-M'03) was born in Haifa, Israel in 1971. He received BSc. and MSc in electrical engineering from the Technion, Israel Institute of Technology in 1996 and 2003, respectively.

He has been a system engineer and technical project manager at Oren Semiconductors from 1996 to 1999 and R&D team leader at Surf Communication Solutions from 1999 to 2001. In 2002 he joined Aelis photonics, Netanya, Israel as manager of signal processing. His interests include digital communications, adaptive signal processing and MIMO systems.

Irving Kalet was born in The Bronx, NY. He received the B.E.E. degree from the City College of New York, New York, in 1962, and the M.S. and Dr.Eng.Sc. degrees from Columbia University, New York, in 1964 and 1969, respectively.

He taught at the City College of New York from 1964 to 1967. He has been living in Israel since 1970. He was a consultant at Bell Laboratories for many years in the fields of HDSL, ADSL, mobile wireless communications, and on the 56 kbit/s modem. He is a Visiting Associate Professor at the Technion, in Israel, for the past six years. He is presently interested in the field of modulation for wireless communications.

Dr. Kalet is a member of Sigma Xi, Tau Beta Pi, and Eta Kappa Nu.

Coupling Efficiency Between Optical Fiber and Ti:LiNbO₃ Channel Waveguide

Marcos Antonio Ruggieri Franco^{1,2}, Laurentino C. de Vasconcellos¹, and José Márcio Machado³

¹Instituto de Estudos Avançados – IEAv/CTA, 12228-840 São José dos Campos–SP, Brazil

²also with Instituto Tecnológico de Aeronáutica – ITA, São José dos Campos –SP, Brazil

³ Depto. de Ciências da Computação e Estatística IBILCE – UNESP, Brazil.

Abstract— The coupling efficiency between a single-mode circular core optical fiber and a single-mode (E_x^x) Ti:LiNbO₃ channel waveguide is calculated as a function of the diffusion process parameters. The waveguide modal analysis was carried out by the finite element method and the fiber to waveguide coupling efficiency was evaluated by the overlap integral factor.

Index Terms— Finite element method, integrated optics, optical fiber coupling, waveguides.

I. INTRODUCTION

The development of low-cost photonic integrated devices for future wide-scale commercialization requires an efficient coupling between the single-mode optical fibers and the optical waveguides as well as a large mechanical alignment tolerance.

Among the several causes of losses in the connection of a fiber and a waveguide, the main causes are the mode field mismatch, the transversal offset, the longitudinal separation, the angular offset, and the refractive index mismatch [1]-[3]. From these, the mode-field mismatch and the relative alignment are the principal sources of fiber-to-waveguide coupling losses. The alignment is typically at least ten times more demanding than the tolerances for integrated circuit packaging. Therefore, the alignment and attachment of fibers are the most difficult challenges facing the optical device packaging.

The analyses of fiber-to-waveguide coupling losses were presented by the several authors [1]-[12]. In the majority of these papers, the coupling efficiency was calculated for z-cut LiNbO₃ substrates using approximating functions to represent the fiber and waveguide mode-field profiles.

In this work, the coupling efficiency between a single-mode circular core optical fiber and a single-mode (E_{11}^x) Ti:LiNbO₃ waveguide was calculated by the overlap integral of the spatial field distribution. For the circular core fiber, the field profile was approximated by the Gaussian function, and for the waveguide formed by diffused Ti-ions on x-cut LiNbO₃ substrate the mode field profile was determined by the finite element method [13]. The coupling efficiency calculations were performed as a function of the initial Ti-strip geometry and the diffusion process parameters. The alignment tolerances in the transversal plane were also determined.

The fiber-to-waveguide attachment scheme and the system reference for the relative alignment are presented in the Fig. 1.

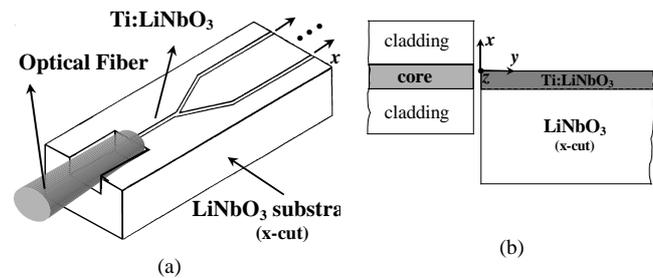


Fig. 1. “End-fire” fiber-to-waveguide coupling. (a) Attached single-mode circular core fiber to Ti:LiNbO₃ waveguide device, (b) Longitudinal view of the relative alignment between fiber and waveguide.

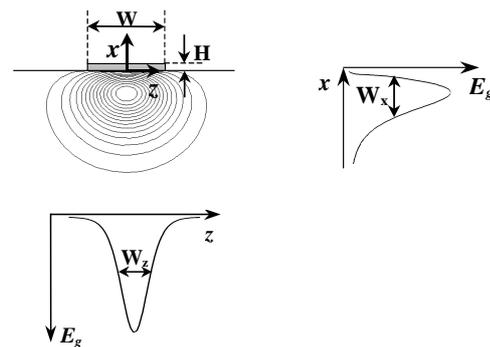


Fig. 2. Initial geometrical parameters of the Ti-strip and guided optical field profile in the diffused Ti:LiNbO₃ waveguide. W_x and W_z are the mode diameter at half maximum of the optical electric field (E_g) intensity.

II. TI:LiNbO₃ CHANNEL WAVEGUIDE

Optical waveguides formed by diffusion of Ti-ions in LiNbO₃ substrate are widely used in the optical integrated circuits. The controllable diffusion parameters determine the mode shape and the confinement. To define the optical channel waveguide, the following manufacturing parameters were taken into account: the initial width (W) and thickness (H) of the Ti-strip, the ion diffusion temperature (T) and the ion diffusion time (t). The calculations were performed for a guide built in an x-cut y-propagation LiNbO₃ substrate for a wavelength of 1.523 μm .

The schematic view of the initial Ti-strip-geometrical parameters, the fundamental mode (E_{11}^x) electric field contour lines and the field-profile in the transversal directions are shown in Fig. 2. Because of the anisotropic

Ti-diffusion characteristic, the mode field presents a nonsymmetrical shape. The W_x and W_z are the mode diameters defined as the full width at half maximum of the field intensity in the transversal directions z and x .

For Ti:LiNbO₃ channel waveguides, the refractive index in the diffused region follows [14]:

$$n_{e,o}^2(x,z,\lambda) = n_{b_{e,o}}^2 + \left[\left(n_{b_{e,o}} + \Delta n_{s_{e,o}} \right)^2 - n_{b_{e,o}}^2 \right] \exp\left(-\frac{x^2}{d_x^2}\right) f\left(\frac{2z}{W}\right) \quad (1)$$

where:

$$f\left(\frac{2z}{W}\right) = \frac{1}{2} \left\{ \operatorname{erf}\left[\frac{W}{2d_z}\left(1 + \frac{2z}{W}\right)\right] + \operatorname{erf}\left[\frac{W}{2d_z}\left(1 - \frac{2z}{W}\right)\right] \right\},$$

e, o denote the extraordinary and ordinary rays respectively, x and z are the coordinates of a point in the substrate, d_z and d_x are the diffusion width and depth respectively, n_b is the substrate refractive index, Δn_s is the variation of the surface index with the wavelength and $f(2z/W)$ represents the solution of the anisotropic diffusion problem.

In addition, $\Delta n_{s_{e,o}}$ is given in terms of H and some fitting parameters [15]:

$$\Delta n_{s_{e,o}}(\lambda) = \left[B_0(\lambda) + B_1(\lambda) \frac{H}{d_{x_{e,o}}} \right] \left(\frac{H}{d_{x_{e,o}}} \right)^{\alpha_{e,o}}, \quad (2)$$

$$\alpha_e = 0.83, \alpha_o = 0.53, \quad 0.6 \leq \lambda (\mu\text{m}) \leq 1.6,$$

$$B_{0e}(\lambda) = 0.385 - 0.430\lambda + 0.171\lambda^2,$$

$$B_{1e}(\lambda) = 9.130 + 3.850\lambda - 2.490\lambda^2,$$

$$B_{0o}(\lambda) = 0.0653 - 0.0315\lambda + 0.0071\lambda^2,$$

$$B_{1o}(\lambda) = 0.4780 + 0.4640\lambda - 0.3480\lambda^2.$$

The diffusion coefficients D_z and D_x , the diffusion width d_z and depth d_x , and the depths of refractive index change profiles d_{xe} and d_{xo} can be calculated by [15]:

$$D_i = D_{i0} \exp\left(-\frac{E_{i0}}{kT}\right), \quad i=x,z \quad (3)$$

$$d_i = 2\sqrt{D_i t}, \quad i=x,z \quad (4)$$

$$d_{x_{e,o}} = \frac{d_x}{\sqrt{\alpha_{e,o}}}, \quad (5)$$

where D_{i0} is the diffusion constant, E_{i0} is the activation energy and k is the Boltzmann constant. These constants for Ti:LiNbO₃ waveguides are presented in Table I [15].

The refractive indexes dispersion of the LiNbO₃ are taken into account by using (6) and (7) with the wavelength (λ) in μm [16], and they are:

$$n_{b_o}^2 = 4.9048 - \frac{0.11768}{0.04750 - \lambda^2} - 0.027169 \lambda^2, \quad (6)$$

$$n_{b_e}^2 = 4.5820 - \frac{0.099169}{0.044432 - \lambda^2} - 0.021950 \lambda^2. \quad (7)$$

III. FINITE ELEMENT MODEL

The modal analysis of the Ti:LiNbO₃ waveguide was carried out by a scalar finite element implementation both for nonhomogeneous and anisotropic media [13]. Accurated optical field profile and propagation constants can be obtained by the application of the finite element method.

TABLE I

COEFFICIENTS OF THE ARRHENIUS LAW FOR Ti:LiNbO₃ GUIDES

D_{z0} ($\mu\text{m}^2/\text{h}$)	5.0 e+9
D_{x0} ($\mu\text{m}^2/\text{h}$)	1.35 e+8
E_{z0} (eV)	2.60
E_{x0} (eV)	2.22

The Helmholtz equation in the scalar approximation, for a lossless, inhomogeneous and anisotropic dielectric optical waveguide with diagonal permittivity tensor, and a harmonically y -propagating wave, can be expressed for the E^x modes as:

$$n_y^2 \frac{\partial}{\partial z} \left(\frac{1}{n_y^2} \right) \frac{\partial (n_z^2 E_z)}{\partial z} + \frac{\partial^2 (n_z^2 E_z)}{\partial z^2} + n_y^2 \frac{\partial^2 E_z}{\partial x^2} + k_0^2 n_z^2 n_y^2 E_z = n_y^2 \beta^2 E_z, \quad (8)$$

and for the E^y modes as follows:

$$\frac{\partial^2 H_z}{\partial z^2} + n_x^2 \frac{\partial}{\partial x} \left(\frac{1}{n_y^2} \frac{\partial H_z}{\partial x} \right) + n_x^2 k_0^2 H_z = \beta^2 H_z, \quad (9)$$

where k_0 is the free space wavenumber and β is the propagation constant. The variables $n_x(z,x)$, $n_z(z,x)$ are the refractive indexes in the transversal directions and $n_y(z,x)$ is the refractive index in the longitudinal direction.

The application of the Weighted Residual Method with the Galerkin Approximation in (8) and (9) yields the matrix equation:

$$[F] \{\phi\}^T = n_{eff}^2 [M] \{\phi\}^T, \quad (10)$$

where $n_{eff} = \beta/k_0$ is the effective index. The matrices for each finite element are given by:

$$[M] = \int_{\Omega} A k_0^2 \{N\}^T \{N\} dz dx, \quad (11)$$

$$[F] = [F_1] - [F_2]. \quad (12)$$

For the E^x modes the matrices are:

$$[F_1] = \int_{\Omega} \left(k_0^2 n_z^2 n_y^2 \{N\}^T \{N\} - n_z^2 \{N\}_z^T \{N\}_z \right) dz dx - \int_{\Omega} n_y^2 \{N\}_x^T \{N\}_x dz dx, \quad (13)$$

$$[F_2] = \int_{\Omega} \left(\delta_z \frac{\partial n_z^2}{\partial z} \{N\}_z^T \{N\} + \delta_y \frac{\partial n_y^2}{\partial x} \{N\}^T \{N\}_x \right) dz dx + \int_{\Omega} \delta_y n_z^2 n_y^2 \frac{\partial g_y^2}{\partial z} \{N\}^T \{N\}_x dz dx. \quad (14)$$

For E^y modes follows:

$$[F_1] = \int_{\Omega} \left[k_0^2 n_x^2 \{N\}^T \{N\} - \{N\}_z^T \{N\}_z \right] dz dx - \int_{\Omega} n_x^2 g_y^2 \{N\}_x^T \{N\}_x dz dx, \quad (15)$$

$$[F_2] = \int_{\Omega} \left[\delta_z g_y^2 \frac{\partial n_x^2}{\partial x} + \delta_y n_x^2 \left(\frac{\partial g_y^2}{\partial x} + g_y^4 \frac{\partial n_y^2}{\partial x} \right) \right] \{N\}^T \{N\}_x dz dx, \quad (16)$$

where $A = n_z^2$, for E^x modes, or $A = 1$, for E^y modes, $g_y = 1/n_y$, $\{N\}_x = \partial\{N\}/\partial x$ and $\{N\}_y = \partial\{N\}/\partial y$ and x and z are the transversal axis and y is the propagation direction. The parameters δ_z and δ_x assume either the value 1 for diffused index in the z and x directions respectively, or zero for constant index. The matrix $[F_2]$ is sparse and nonsymmetrical because of the presence of terms with refractive index partial derivative.

In this work, the variation of the refractive indexes and their spatial derivatives inside each finite element are expanded in terms of the base functions $\{N\}$:

$$n_i^2 = \{N\} \{n_i^2\}^T, \quad i, j = x, y, z$$

$$\frac{\partial n_i^2}{\partial j} = \{N\} \left\{ \frac{\partial n_i^2}{\partial j} \right\}, \quad \frac{\partial g_y^2}{\partial j} = \{N\} \left\{ \frac{\partial g_y^2}{\partial j} \right\}$$

The finite element formulation was implemented in a personal computer (Athlon 900 MHz, 512 Mb RAM) using the Matlab software. For a single modal analysis a typical CPU time is about 10 minutes. The geometric models, the attribution of physical properties and boundary conditions, the finite element unstructured mesh generation (using Delaunay algorithm) and graphic visualization of the electromagnetic fields were done using the LevSoft, which was developed in the Institute of Advanced Studies – IEAv/CTA for finite element applications [17].

IV. COUPLING EFFICIENCY CALCULATIONS

The coupling efficiency between a single-mode circular core optical fiber and a single-mode Ti:LiNbO₃ waveguide was calculated by the overlap integral of the

spatial field distribution:

$$\Psi = \frac{\left| \iint E_f E_g dz dx \right|^2}{\iint |E_f|^2 dz dx \iint |E_g|^2 dz dx}, \quad (17)$$

where E_f and E_g are the electric fields in the fiber and in the optical waveguide, respectively.

For the circular core fiber, the field profile is well approximated by the Gaussian function [4]-[9]:

$$E_f(r) \propto \exp(-r^2/2a^2) \quad (18)$$

where $r^2 = x^2 + z^2$ and a is the mode radius of the fiber at the half-width-at-half-maximum field intensity.

Fig. 3 presents the fiber to waveguide coupling efficiency and the waveguide mode diameter in the transversal directions (W_x, W_z) as a function of the initial width of Ti-strip for different diffusion times. The following set of fixed parameters was assumed: $H=80$ nm, $T=1050$ °C, $\lambda=1.523$ μ m and the fiber mode diameter $2a=5$ μ m. For this particular case, the waveguide is single-mode for Ti-strip widths smaller than 7 μ m for $t=3$ h and smaller than 10 μ m for $t=12$ h. The coupling efficiency values reach approximately 92% on the single-mode region. However, for long diffusion times the maximum coupling efficiency condition is near the cutoff second mode.

The coupling efficiency and the mode diameters as a function of the initial Ti-strip thickness (H) and diffusion time are presented in Fig. 4 for $W=5$ μ m, $T=1050$ °C, $\lambda=1.523$ μ m and $2a=5$ μ m. When $t=3$ h, the waveguide is single-mode for $H \leq 95$ nm and the cutoff frequency for the fundamental mode is reached for $H \leq 55$ nm. For the case of $t \geq 6$ h, the propagation is only single-mode but the cutoff frequency changes to higher Ti thickness. In all cases the maximum coupling efficiency is approximately 92 %, however for $t=6$ h the efficiency values presents low sensitivity to the thickness variations, allowing to simplify the Ti-strip deposition requirements.

Fig. 5 shows the coupling efficiency and the waveguide mode diameters as functions of both diffusion temperature and the diffusion time for $W=5$ μ m, $H=80$ nm, $\lambda=1.523$ μ m, and $2a=5$ μ m. Again, the maximum coupling efficiency is about 92 % for each diffusion time considered. The maximum efficiency values are reached for temperatures between 1000°C and 1100°C and correspond to waveguide mode diameters near to 5 μ m which is the mode diameter in the fiber.

Figs. 3-5 show the maximum values of coupling efficiency for different fabrication parameters. Even though, the related fiber positions are essential to define the geometrical parameters of the alignment grooves, which are manufactured either by chemical etching or laser ablation system. The Figs. 6(a)-6(c) present the center fiber position ($z=0$ μ m, fixed) corresponding respectively to the data set of the Figs. 3-5. The simulations show that the fiber position changes around 1 μ m with W or H variations. For temperature variations between 900°C and 1150°C the fiber position must be modified by ≈ 5 μ m.

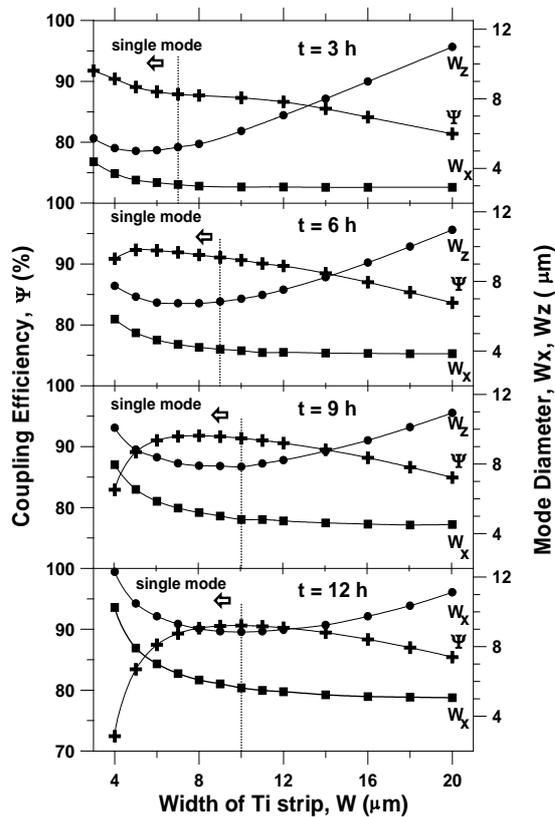


Fig. 3. The coupling efficiency and waveguide mode diameter as functions of the initial width of Ti-strip and the diffusion time.

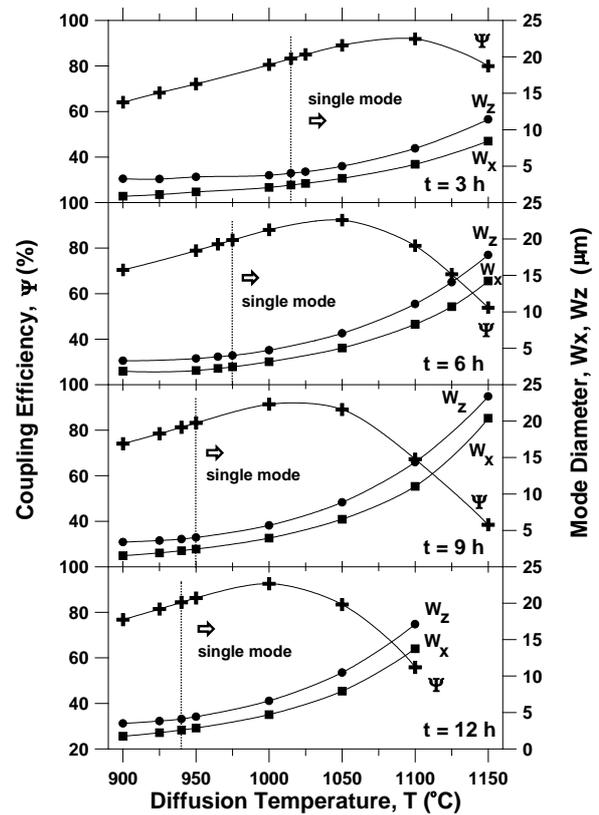


Fig. 5. The coupling efficiency and waveguide mode diameter as functions of the diffusion temperature and the diffusion time.

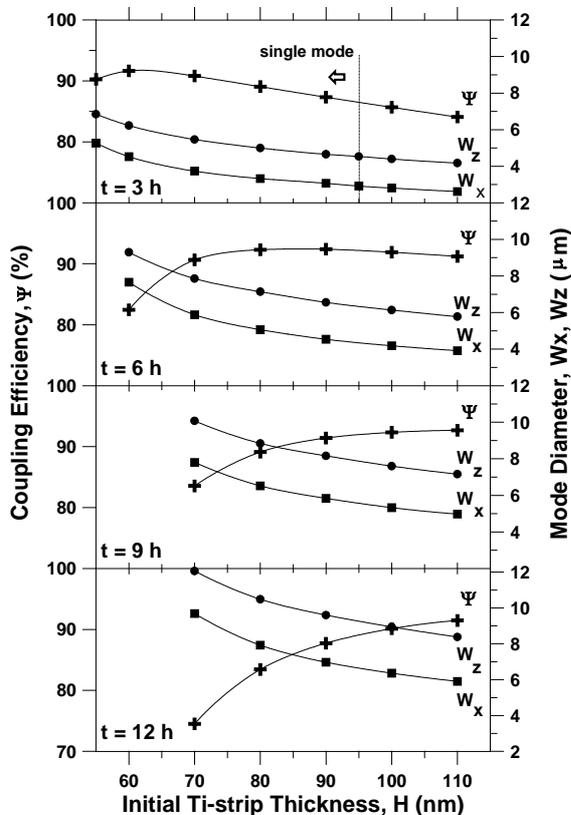


Fig. 4. The coupling efficiency and waveguide mode diameter as functions of the initial Ti-strip thickness and the diffusion time.

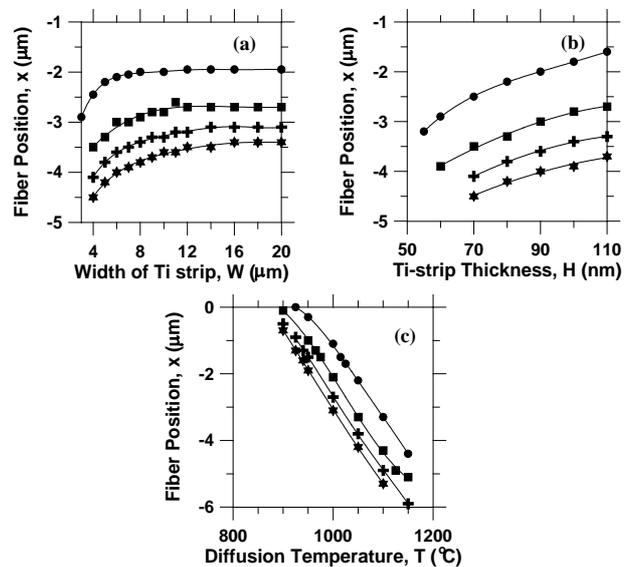


Fig. 6. The vertical center fiber position, x ($z = 0 \mu\text{m}$), corresponding to the maximum coupling efficiency as a function of the waveguide fabrication parameters. $\lambda = 1.523 \mu\text{m}$. \circ $t = 3 \text{ h}$, \square $t = 6 \text{ h}$, \star $t = 9 \text{ h}$, and \times $t = 12 \text{ h}$.

The alignment tolerances studies are illustrated in Fig. 7 in which the coupling efficiency is plotted against transversal offsets for the following fixed parameters: $2a = 5 \mu\text{m}$, $W = 5 \mu\text{m}$, $H = 80 \text{ nm}$, $T = 1050^\circ\text{C}$, and $\lambda = 1.523 \mu\text{m}$. The mechanical tolerances were determined considering a maximum connection loss equal -1.0 dB ($\approx 20.6 \%$).

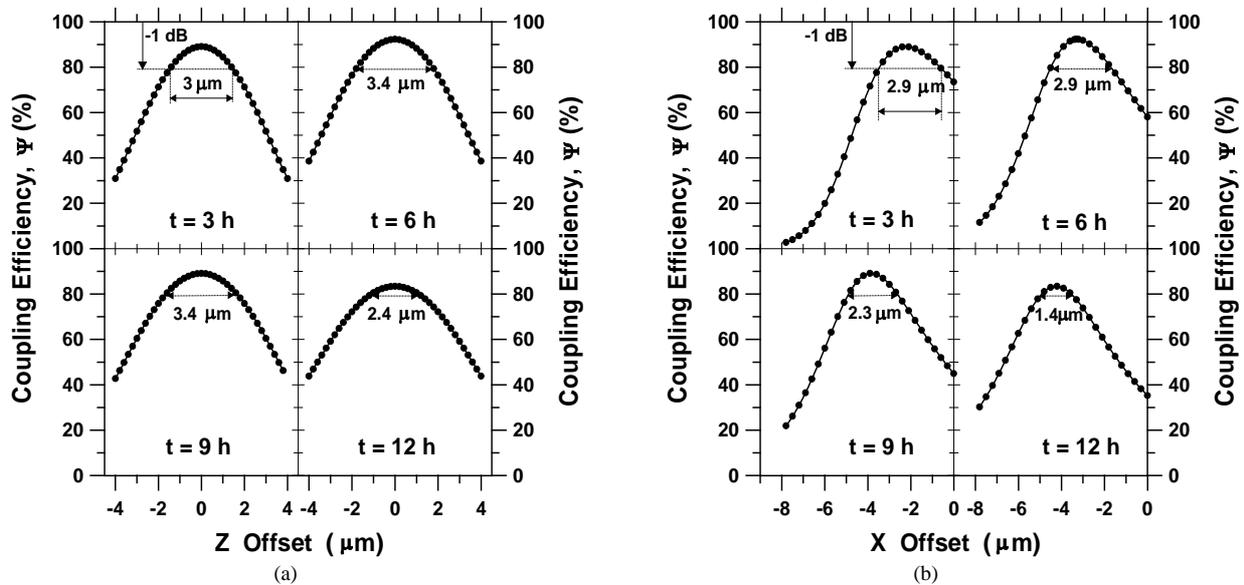


Fig. 7. Coupling efficiency as a function of the transversal offset. (a) Offset in z-direction, (b) offset in x-direction.

TABLE II
COUPLING EFFICIENCY FOR COMMERCIAL OPTICAL FIBERS

λ (μm)	$2a$ (μm) fiber	n_{eff} guide	W_z (μm) guide	W_x (μm) guide	Ψ (%)
0.820	3.748	2.174061	4.28	2.758	88.517
1.060	5.092	2.159105	5.20	3.444	86.830
1.300	5.656	2.147652	6.14	4.215	89.234
1.550	6.718	2.138914	7.22	5.145	89.792

The results shown in the Figs. 3-7 were calculated for a typical fiber mode diameter ($2a=5\mu\text{m}$) and the waveguide mode dimensions were calculated for $\lambda=1.523\mu\text{m}$. By taking into account some commercial fiber characteristics is possible to obtain the attachment condition for special cases. Table II presents an example for four commercial fibers (manufactured by 3M, PM series) where the following fixed parameters were considered $W=5\mu\text{m}$, $H=80\text{nm}$, $T=1050^\circ\text{C}$, and $t=6\text{h}$.

I. CONCLUSIONS

The fiber-to-diffused waveguide coupling efficiency was calculated by using the finite element modal analysis for the Ti:LiNbO₃ waveguide. The effect of waveguide fabrication parameters on the coupling was verified. The relative fiber positions to obtain the maximum coupling efficiency were determined. This study permits the design of groove channels to attach and align the fiber-waveguide connection. The mechanical precision alignment requirements were determined by analyzing the fiber transversal offsets.

ACKNOWLEDGMENT

The authors wish to acknowledge the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) by the partial support, process number 98/07789-7.

REFERENCES

- [1] L. Eldada and J. T. Yardley, "Modal Analysis for Optimization of Single-Mode Waveguide Pigtailling and Fiber Splicing," *Appl. Opt.*, vol. 37, N. 33, pp.7747-7751, Nov. 1998.
- [2] M. L. Tuma and G. Beheim, "Calculated Coupling Efficiency Between an Elliptical Fibres and a Silicon Oxynitride RIB Waveguide," *NASA Tech. Memo.* 106850, 1995.
- [3] N. H. Zhu, X. S. Zheng, Y. K. Lin, and X. Yu, "Calculation of the field distribution of a Ti:LiNbO₃ Optical Waveguide and its Applications," *Optical and Quantum Electronics*, vol. 24, pp.737-743, 1992.
- [4] S. Y. Wang and S. H. Lin, "High Speed III-V Electrooptic Waveguide Modulators at $\lambda = 1.3\mu\text{m}$," *J. Lightwave Technol.*, vol. 6, N. 6, pp.758-771, Jun. 1988.
- [5] E. J. Murphy, "Fiber Attachment for Guided Wave Devices," *Journal of Lightwave Technol.*, vol. 6, N. 6, pp.862-871, Jun. 1988.
- [6] P. G. Suchoski Jr. and R. V. Ramaswamy, "Minimum-Mode-Size Low-Loss Ti:LiNbO₃ Channel Waveguides for Efficient Modulator Operation at $1.3\mu\text{m}$," *IEEE Journal of Quantum Electronics*, vol.23, N. 10, pp. 1673-1679, Oct. 1987.
- [7] J. J. Veselka and S. K. Korotky, "Optimization of Ti:LiNbO₃ Optical Waveguides and Directional Coupler Switches for $1.56\mu\text{m}$ Wavelength," *IEEE Journal of Quantum Electronics*, vol. 22, N. 6, pp.933-938, Jun. 1986.
- [8] L. McCaughan and E. J. Murphy, "Influence of Temperature and Initial Titanium Dimensions of Fiber-Ti:LiNbO₃ Waveguide Insertion Loss at $\lambda = 1.3\mu\text{m}$," *IEEE Journal of Quantum Electronics*, vol. 19, N. 2, pp.131-136, Feb. 1983.
- [9] R. C. Alferness, V. R. Ramaswamy, S. K. Korotky, M. D. Divino, and L.L. Buhl, "Efficient Single-Mode fiber to Titanium Diffused Lithium Niobate

- Waveguide Coupling for $\lambda = 1.32 \mu\text{m}$,” *IEEE Journal of Quantum Electronics*, vol. 18, N. 10, pp. 1807-1813, Oct. 1982.
- [10] M. Fukuma and J. Noda, “Optical Properties of Titanium-Diffused LiNbO₃ Strip Waveguides and Their Coupling-to-a Fiber Characteristics,” *Applied Optics*, vol. 19, N. 4, pp.591-597, Feb. 1980.
- [11] C. H. Bulmer, S. K. Sheem, R. P. Moeller, and W. K. Burns, “High-Efficient Flip-Chip Coupling Between Single-Mode Fibers and LiNbO₃ Channel Waveguides,” *Applied Physics Letters*, vol.37, N. 4, pp.351-353, Aug. 1980.
- [12] W. K. Burns and G. B. Hocker, “End Fire Coupling Between Optical Fibers and Diffused Channel Waveguides,” *Applied Optics*, vol. 16, N. 8, pp. 2048-2050, Aug. 1977.
- [13] M. A. R. Franco, A. Passaro, J. R. Cardoso, and J. M. Machado, “Finite Element Analysis of Anisotropic Optical Waveguide with Arbitrary Index profile,” *IEEE Trans. on Magnetics*, vol. 35, pp. 1546-1549, May. 1999.
- [14] D. Zhang, C. Chen, G. Ding, J. Zhang, and Y. Cui, “Dependence of Ti-Diffused Er:LiNbO₃ Laser Efficiency on Waveguide Fabrication Parameters and Pump Wavelength,” *IEEE Journal of Quantum Electronics*, vol. 33, N. 7, pp.1231-1235, Jul. 1997.
- [15] S. Fouchet, A. Carencó, C. Daguét, R. Guglielmi, and L. Riviere, “Wavelength Dispersion of Ti Induced Refractive Index Change in LiNbO₃ as a Function of Diffusion Parameters,” *Journal of Lightwave Technol.*, vol. 5, N. 5, pp. 700-708, May. 1987.
- [16] D. S. Smith, H. D. Riccius, and R. P. Edwin, “Refractive Indices of Lithium Niobate,” *Opt. Commun.*, vol. 17, N. 3, pp. 332-335, Jun. 1976.
- [17] N. M. Abe, A. Passaro, M. A. R. Franco, F. Sircilli, V. A. Serrão, D. H. Odan and F. J. R. Santos, “Um Sistema de Software para Análise de Dispositivos e Componentes de Óptica Integrada, Fibras Ópticas e Microondas,” Proceedings of the V Congresso Brasileiro de Eletromagnetismo - CBMag 2002, Gramado – RS, November 04 a 06, Brazil, 2002.

Marcos Antonio Ruggieri Franco was graduated in Physics at Pontifícia Universidade Católica de São Paulo-Brazil (PUC-SP), in 1983. In 1991, he finished his Master Science degree in Nuclear Physics at Instituto de Física da Universidade de São Paulo-Brazil (IFUSP). In 1999, he received his doctorate in Electrical Engineering from the Escola Politécnica da Universidade de São Paulo-Brazil (POLI-USP). In 1987, he joined the research team of the Division of Applied Physics at Institute of Advanced Studies (IEAv) at Centro Técnico Aeroespacial (CTA). Since 2001, he is also associated professor of the pos-graduate course of Electronic and Computation Engineering from the Instituto Tecnológico de Aeronáutica (ITA). His major areas of interest are the application of the Finite Element Method for the design of electromagnetic devices such as microwave and optical waveguides, fiber optics, integrated optics, antennas and electromagnetic scattering problems.

(email: marcos@ieav.cta.br)

Laurentino Corrêa de Vasconcellos Neto, was graduated in Physical Science and Industrial Mechanical Engineering and pos-graduated in Optical Engineering. His Master Science was in Biomedical Engineering and his Doctorate was in Electrical Engineering obtained from the Escola Politécnica da Universidade de São Paulo-Brazil (POLI-USP). Since 1975, he is working at the Centro Técnico Aeroespacial as a researcher. Since 2001, he is associated professor of the pos-graduated course of Bioengineering at Universidade do Vale do Paraíba-UNIVAP, Brazil.

(email: vasko@ieav.cta.br)

José Márcio Machado obtained his Doctorate in Electrical Engineering from the Escola Politécnica da Universidade de São Paulo (1993). In 1985, he finished his Master Science at the Centro Brasileiro de Pesquisas Físicas (CBPF). Nowadays, he is professor of Mathematics at the Universidade do Estado de São Paulo–Brazil (UNESP-São José do Rio Preto) at Science Computation. His main researches interests are numerical methods, differential equations and applications for Physics, Mathematics and Engineering.

(email: jmarcio@dce.ibilce.unesp.br)

The Development of a Submm-Wave Uncooled Bolometric System and Field Test

Roberto R. Neli, Arline M. Melo, Cristiano J. N. Arbex, Maria B. P. Zakia, Ioshiaki Doi, Pierre Kaufmann, Jacobus W. Swart and Edson Moschim.

Abstract - Developments directed to potential interest for telecommunications in the submillimeter and far infrared wavelengths band have been increasing in recent years. The detection of radiation in that spectral region has various other important applications in remote passive radiometric sensing, some times called as thermosensing, such as imaging of field targets, power line surveys, night vision, radio meteorology, atmospheric propagation and astrophysics. The development of a non-imaging bolometer detector for incoherent radiation in the submillimeter bands is presented. Construction details are presented for a detector's active material using polycrystalline and amorphous silicon. We describe the bandpass filters to be placed in front of the detector, which are being designed to be constructed using the resonant mesh technology, centered at the peak frequencies around 405, 670 and 850 GHz. The bolometric system will be submitted to qualification and performance tests at the focal plane of a 1.5-m submillimeter reflector located at a high altitude dry site (El Leoncito, Argentina Andes), using the Sun as a source of blackbody radiation, and a beacon transmitter at 405 GHz located in the far field with respect to the reflector. The first applications considered for the system are measurements of atmospheric propagation, solar imaging and tentative detection of solar transients (flares) in the selected bands.

Resumo - Nos últimos anos tem se dado crescente importância ao desenvolvimento de microbolômetros, atendendo interesses potenciais para telecomunicações, da faixa submilimétrica ao infravermelho distante. A detecção de radiação nesta

Roberto R. Neli is a doctoral graduate student at CCS – FEEC, Unicamp, Campinas SP, Brazil.

Arline M. Melo is a MS graduate student at CCS – FEEC, Unicamp, Campinas SP, Brazil and stays part time at CRAAM – Universidade Presbiteriana Mackenzie, São Paulo, SP, Brazil;

Cristiano J. N. Arbex is a MS graduate student at CCS – FEEC, Unicamp, Campinas SP, Brazil;

Maria Beny is full-time researcher at CCS – Unicamp, Campinas, SP, Brazil;

Ioshiaki Doi, is a professor at FEEC-Unicamp and Vice-Director of CCS – Unicamp, Campinas, SP, Brazil;

Pierre Kaufmann is professor at CRAAM - Universidade Presbiteriana Mackenzie, São Paulo, SP, Brazil and part-time researcher at CCS – Unicamp, Campinas, SP, Brazil;

Jacobus W. Swart is professor at FEEC and Director of CCS – Unicamp, Campinas, SP, Brazil;

Edson Moschim is professor at FEEC, Unicamp, Campinas, SP, Brazil;

This research is being partially supported by FAPESP, Brazil, contracts 99/06126-7 and 02/04774-6, CNPq contracts 690190/02-6 and 140583/02-5, CAPES for CJNA master course fellowship, and by

CONICET.

região espectral tem várias outras importantes aplicações em sensoriamento passivo remoto, também chamado de termosensoriamento, tais como imageamento de alvos no campo, supervisão de redes elétricas de potência, visão noturna, rádio meteorologia, propagação atmosférica e astrofísica. Este trabalho apresenta o desenvolvimento de um detector bolométrico, não imageador, de radiação incoerente em bandas submilimétricas, bem como os testes de campo previstos. São apresentados detalhes de construção do material ativo, formado por silício policristalino ou amorfo, do detector utilizando.

Descrevemos os filtros passa-banda a serem instalados frente o coletor de fótons e o detector, projetados e construídos para frequências centradas em 405, 670 e 850 GHz. O sistema bolométrico será submetido a testes de qualificação e de desempenho no plano focal de um refletor submilimétrico de 1,5m localizado em sítio seco de elevada altitude (El Leoncito, Andes Argentinos), usando o Sol como fonte de radiação com característica de corpo negro, e um transmissor artificial de 405 GHz localizado à distância remota com relação ao refletor. Como primeiras aplicações consideradas para o sistema estão medidas de propagação atmosférica, imageamento solar, e detecção tentativa de transientes solares (explosões) nas bandas

Index Terms—Infrared detectors, submillimeter transmission, silicon devices, bandpass filters, bolometers.

I. INTRODUCTION

The detection of non-coherent electromagnetic waves in the range that includes millimetric, submillimetric waves until the far-infrared, uses bolometers or microbolometer array sensors which have their electric characteristics changed by the incidence of electromagnetic radiation.

In this project we will use resistive bolometers, where their electric resistance modifies when the radiation is absorbed. Their basic structure consists of a fine film of metal or semiconductor with an absorber film placed above, usually suspended by a substrate producing a thermal isolation (Fig. 1).

The detectors used in the submm-wave – IR range can be cooled or not. In more recent years uncooled detectors have been widely used. They provide good sensitivity at room temperature bringing advantages like low cost, low weight and low power consumption.

Among the various applications for the detectors that operate in these high frequencies there is an important potential future for telecommunications, from the ground in the range where there are atmospheric “windows” for transmission and above the Earth’s troposphere for communications between platforms (balloons, airplanes and stationary devices) or between satellites.

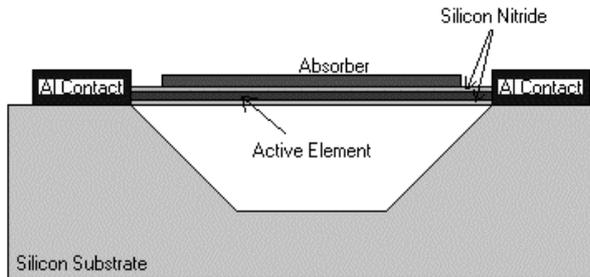


Fig. 1. Microbolometer Structure.

Other important applications to be mentioned are the passive imaging made through clouds, fog, rain, smoke and even through walls; devices that allow night vision; terrestrial mining detection; fire detection; security devices; cameras for medical and industrial applications; mine sensors in battlefields for military purposes. Several thermosensing applications have been developed for agriculture, ground and crops surveys [Refs. 1-3].

This range of the electromagnetic spectrum is also of a great interest for space sciences, because it brings essential spectral information on the cosmic background radiation, on distant galaxies recently formed and on the initial phases of stellar formation in gas clouds in our own galaxy [4].

The diagnostics of astrophysical objects in the submm-IR range – with emphasis to the Sun - not only in the continuum but also in spectral lines; are in the front line of current research. In this range of wavelengths it is possible to distinguish thermal from the non-thermal emissions caused by relativistic particles. These measurements can be developed from the ground, or from space platforms in balloons and satellites.

The design of a room-temperature bolometric detection system has been developed, which basic technical features, qualification tests, and first applications are described here.

II. DEPOSITION AND PROCESSING PARAMETERS

An alternative way to perform thermally insulated bolometers consists of micromachining the active element to form a microbridge which is anchored to the silicon substrate through thin supports. The thermal detectors are based on the effect of temperature change caused by the heat absorbed from incident radiation upon the detector [5]. They have a number of advantages including wide spectral response, room temperature operation and low cost. The thermal insulation is determined by the geometry and the thermal conductivity of the supports. Strong candidates for the active material to be used are poly and amorphous silicon. Their electrical temperature

coefficient of resistance and conductivity (TCR) can be easily controlled by boron doping. It is fully compatible with integrated circuit (IC) technology, can operate in a wide range of temperature and presents low cost technology.

Using silicon anisotropic etching in potassium hydroxide solution (KOH, 10M at 82°C), various shapes of bridges were fabricated as shown in Fig. 2.

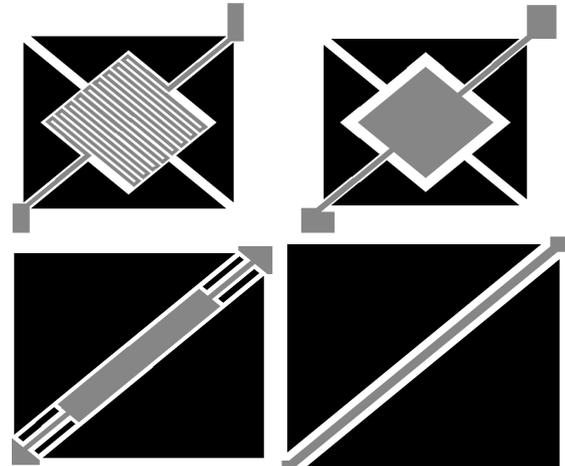


Fig. 2. Shapes of Bridges.

Nitride silicon or silicon oxide films are prepared on a (100)-oriented silicon substrate as support films acting as microbridges. First, the silicon wafer has been covered with a 500nm silicon nitride deposited by electron synchrotron resonance (ECR), or with a 1µm silicon oxide, grown in a wet oxidation. In the following step, a 1µm thick poly or amorphous silicon layer is deposited on top of silicon nitride using chemical vapor deposition (CVD). The silicon nitride layer is implanted with a low boron dose ($3 \times 10^{13} \text{ B cm}^{-2}$) in order to obtain the desired resistivity and TCR. A 100 nm thick layer of silicon nitride is deposited onto poly silicon in order to insulate the submm-wave/infra-red absorber from the active element [6]. These three layers consist the pattern in order to define the support legs and active area, whereas the support legs are highly doped ($10^{16} \text{ B cm}^{-1}$) [7], to act as electrical contacts. This highly doping also forms a *p-n* junction with silicon substrate proving an electrical insulation. After deposition of metal contacts, the processing of the absorber formed by a porous gold layer.

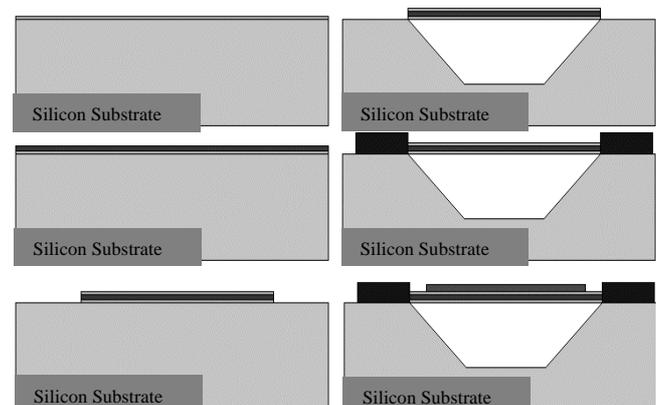


Fig. 3. Schematic flow chart of silicon microbolometers

fabrication.

This layer is known as “gold black” or “gold soot” and is obtained by gold evaporation in nitrogen atmosphere. [8]. The absorber function is to convert the incident radiation into heat. It must have high absorption efficiency, high reproducibility and must be compatible with standard process. We can observe a bolometer fabrication scheme in Fig. 3.

III. RESONANT MESH AS BANDPASS FILTERS

There are many applications for bandpass filters in the submm-wave to far-infrared range of frequencies. For example, “tunnel” junction mixers need bandpass filters to avoid the saturation by thermal radiation [9]. The noise equivalent power of a bolometer can also be reduced by using a bandpass filter to block the thermal radiation over the spectral range of the detector [9].

Bandpass filters are used in bolometric systems to select spectral frequencies of non-coherent electromagnetic waves in the range that includes millimetric, submillimetric extending to the far-infrared waves, on various applications in passive sensing or imaging. In the present case, the filters designs must have center frequencies to fit the atmospheric transmission windows, similarly to the selections made by Chase and Joseph [10]. Such filters should present good transmission at the center of the band and excellent blocking for lower and higher frequencies.

There is a variety of resonant mesh structures with different shapes that produce bandpass response, such as arrays of crosses; arrays of circular and square conducting rings and pairs of concentric circular conducting rings [Ref. 9-12]. Our attention will be directed to arrangements of cross-shaped elements.

It is known that a conductive plate with a square opening acts as a high-pass filter, and in the transmission lines model context is considered as an inductive arrangement [10,11].

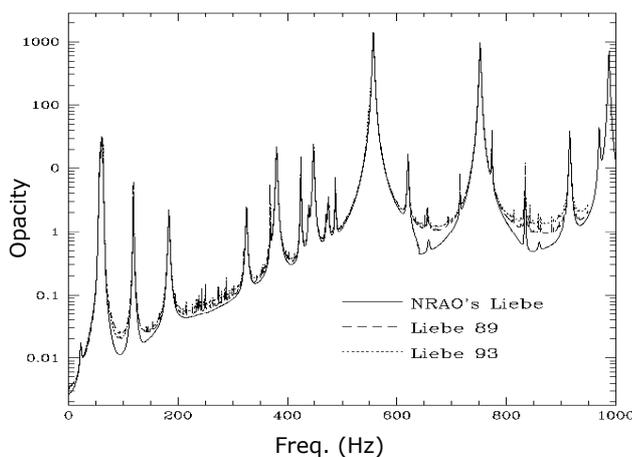


Fig. 4. Atmospheric opacity (in nepers) vs. frequency (in GHz).

Similarly, an arrangement with a complementary structure, acts as a low-pass filter also called as a capacitive arrangement.

Associating those two arrangements, a bandpass behavior is obtained, as is the case of the cross-shaped elements arrangement [11]. Considering typical atmospheric spectral transmission characteristic [Refs. 13-16], we selected transmission “windows” centered in the frequencies around 405, 670 e 850 GHz for the filters.

These frequencies correspond, to the following wavelengths: 741 μm, 448 μm and 352 μm, respectively as we can see in Fig. 4. The design parameters were studied for bandpass filters centered on those bands.

Several authors have used cross-shaped elements as bandpass filters with different notations to characterize the crosses' dimensions [Refs. 9-11]. In this work we will follow the nomenclature given by Chase and Joseph [10]. The cross-shaped elements have the design shapes shown in Fig. 5, where g is the periodicity, L is the cross length, $2b$ is the cross width and $2a$ is the separation between crosses.

$$g = L + 2a \tag{1}$$

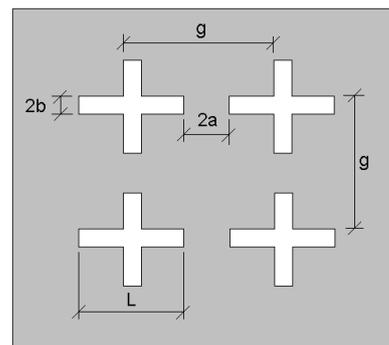


Fig. 5. Definition of the cross-shaped mesh parameters.

The resonance wavelength (wavelength where occurs the maximum transmission) is given by [10,12]:

$$\lambda_0 = 2.LL \tag{2}$$

on the condition that the b/a ratio is smaller than 1.

Thus, for each value of the filter central frequency, there are corresponding values for L , close to a half-wavelength.

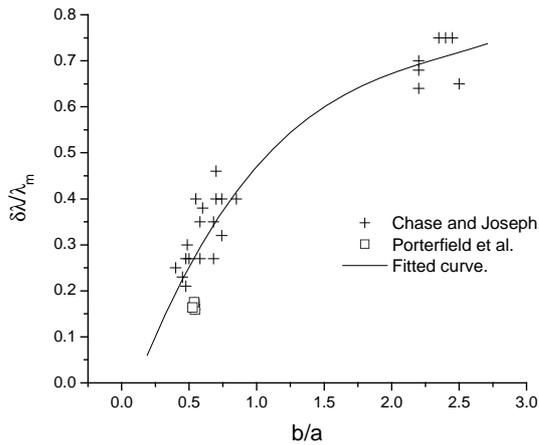


Fig. 6. Measured data and fitted curve for the bandwidth vs b/a ratio.

For 220 GHz, we obtain $L = 649 \mu\text{m}$. For 410 GHz, $L = 348 \mu\text{m}$ and for 670 GHz, $L = 213 \mu\text{m}$. Our design looks for a small bandpass, nearly 25% of the central wavelength. Fig. 6 shows the dependence of the bandwidth with the b/a ratio. The experimental data were obtained from Porterfield et al. [9] and Chase and Joseph [10]. Fitting the data of Fig. 6 derived an empirical equation relating the bandpass and the b/a ratio:

$$\frac{\delta\lambda}{\lambda_m} = -0.0839 + 0.8115\left(\frac{b}{a}\right) - 0.2984\left(\frac{b}{a}\right)^2 + 0,0409\left(\frac{b}{a}\right)^3 \quad (3)$$

where using $\delta\lambda/\lambda_m = 0.25$, we get $b/a \cong 0.5$.

IV. THE CONSTRUCTION OF FILTERS

Using the Eq.1 and Eq.2, adopting g equal 80% of the resonant wavelength $g = 0.8 \lambda_0$ and $b/a \cong 0.5$ as the result of the Eq.3, the following filter parameters are obtained:

f (GHz)	g (μm)	L (μm)	a (μm)	b (μm)
220	1091	649	221	111
410	586	348	119	60
670	358	213	73	37

The filter construction is being considered with the use of photolithography process associated to an etching process or metal deposition. Porterfield et al. [9] uses the copper electroplating process for the crosses formation. Initially a thin copper layer is deposited on a glass substrate. Then, photoresist is applied on the whole copper surface which is submitted to exposure to UV followed by the photoresist development, so that the crosses become in high relief. Then the copper electroplating around the photoresist crosses is done. Removing the photoresist produces the crosses cavity. The initial copper layer is removed with a light etching.

Chase and Joseph [10] and Tomaselli et al. [11] used the etching process to obtain the mesh crosses. In this process a metal substrate (aluminum [12] or copper [11])

is deposited onto a thin Mylar substrate. After the photoresist is applied the exposure is done. Later, the metal etching is accomplished to obtain opaque crosses.

MacDonald et al. [12] uses the liftoff process. A polyester substrate is glued with photoresist on a base formed by a silicon substrate. The polyester is covered with photoresist to perform the exposure and development, so that the crosses become in high relief. A gold or lead layer is later deposited by sputtering, so that the cross is surrounded by metal. Acetone can be used to remove the polyester of the substrate.

For the present design we will consider the processes described by Porterfield et al. [9], Chase and Joseph [10] and Tomaselli et al. [11].

Fig. 7 illustrates two enlarged views of a 565 GHz mesh filter built in the CCS laboratory for bandpass testing purposes.

The complete bolometric system will be assembled at the end of a “photon trap” to achieve maximum collection of radiation using the technique of non-imaging concentrator, also known as Winston cone [17] [18].

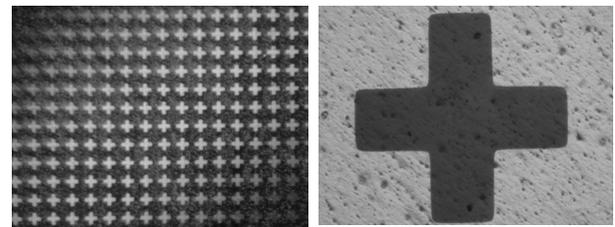


Fig.7. Enlarged views of resonant mesh filter designed for center frequency of 565 GHz, built on aluminum/plastic film at CCS laboratory.

V. OBJECTIVES AND TESTS

This project proposes the establishment of definitions, prototype construction and installation of a bolometric system in the focal plane of a 1.5-m reflector for submm-waves (SST project) installed in Argentina Andes [19] – which was made available for the present research. The assembly will be used

Fraction of observing time

Daytime atmospheric opacity at 212

Daytime atmospheric opacity at 405

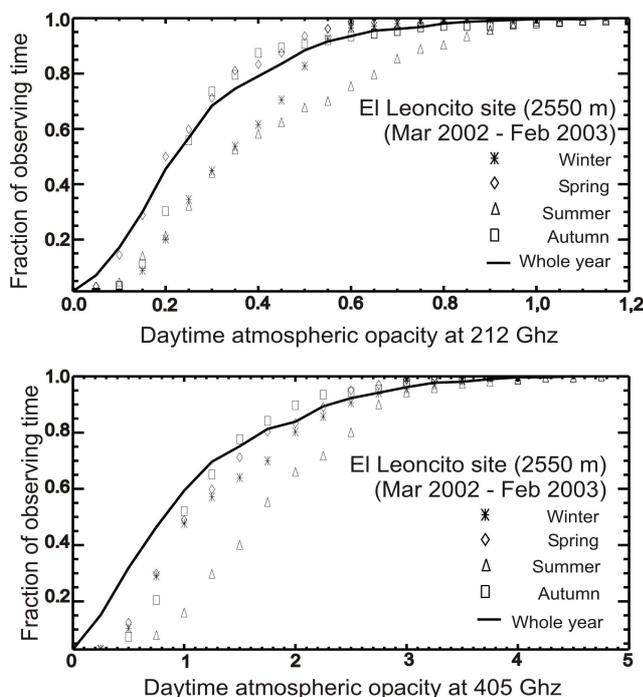


Fig. 8. The cumulative opacity plots for El Leoncito for 212 GHz (top) and 405 GHz (bottom) for one year of measurements [17,18].

for tests on its performance, radiation detection, and qualification. Solar emission will be used as a source of continuum radiation, exhibiting a black body spectrum. At one frequency band, 0.4 THz, a beacon transmitter will be used at the same site, located in the far field with respect to the reflector.

Recent measurements of atmospheric opacity for the site of El Leoncito show that nearly 60% of the observing time has opacities smaller than 0.35 (Nepers) for 212 GHz and 1.5 (Nepers) for 405 GHz, showed in Fig. 8. This study was made for measurements during one year of observations (Mar 2002 -Feb 2003) [20] [21]. According to model predictions [Refs. 13-16] the fraction of observing time for an atmospheric opacity smaller than 2.0 (Nepers) at 670 GHz and 850 GHz at El Leoncito is expected to be in the range of 50 - 70 %, therefore nearly 170 days in a year.

A number of applications are also being considered for this initial phase. They will concentrate on atmosphere propagation, which inhomogeneities and opacity determinations are not very well known in this range of frequencies [13], as well as on solar imaging and possible solar transients detections (flares). These measurements will complement the works that are been done at submm-waves propagation using SST at frequencies for which there are coherent heterodyne radiometers available (212 and 405 THz).

VI. FINAL REMARKS

It has been shown that submm-wave-far IR uncooled sensors using resistive bolometers can be used to perform high frequency measurements of uncoherent radiation in

various applications. A small bias is generated when submm-infrared radiation from the emitting source is directed to the detector, measured by an external circuit. Poly and amorphous silicon can be used as active element built in different patterns which will be measured to observe the best result. The resistivity and TCR of the active element can be obtained by ion implantation and low-temperature annealing. The microbolometer was coupled to a submm-IR absorber with low thermal mass, forming in this way an efficient bolometer. Bandpass filters using resonant mesh technology will be used to limit the operation frequency band, defining the spectral resolution of the measurements. We selected atmospheric transmission windows to centralize the frequencies 400, 670 and 850 GHz for the filters designs. The complete process for their fabrication was described. Performance and qualification tests will be performed placing the bolometer system in the focal plane of a 1.5-m submm-wave reflector, located in a high and dry mountain site (El Leoncito, San Juan, Argentina). Solar black body emission will be used as a test radiation source for the three bands, together with a 410 GHz far field beacon transmitter at the same site. Initial applications will be directed to atmospheric transmission measurements, solar imaging and transient detection.

REFERENCES

- [1] D. S. Tezcan, T. Akin, "A CMOS N-well Microbolometer FPA with Temperature Coefficient Enhancement Circuitry," SPIE 4369, 240-249, 2001.
- [2] D. Murphy, M. Ray, R. Wyles, J. Asbrock, N. Lum, A. Kennedy, J. Wyles, C. Hewitt, G. Graham, W. Radford, J. Anderson, D. Bradley, R. Chin, T. Kostrzewa, "High sensitivity (25 μm pitch) microbolometer FPAs and application development", SPIE 4369, 223-234, 2001.
- [3] O. Savry, P. Agnès, "Arrays of bolometer detector for millimeter wave imaging: feasibility, performances and construction of a prototype," Présenté à IRMMW Toulouse, 2001.
- [4] T. G. Phillips, "Techniques of submillimeter astronomy", Millimetre and Submillimetre Astronomy, vol. 147, pp. 1-25, 1988.
- [5] L. Dobrzanski, E. Nossarzewska-Orlowska, Z. Nowak, J. Piotrowski, "Micromachined silicon bolometers as detectors of soft X-ray, ultraviolet, visible and infrared radiation," Sensors and Actuators A, vol. 60, pp. 154-159, 1997.
- [6] S. Sedky, P. Fiorini, M. Cayamax, A. Verbist and C. Baert, "IR bolometers made of polycrystalline silicon germanium," Sensors and Actuators A, vol. 66, pp. 193-199, 1998.
- [7] T. A. Core, W. K. Tsang, S. J. Sherman, "Fabrication technology for an integrated surface-micromachined sensor," Solid State Technology, vol. 36, 1993.
- [8] S. J. Lee, Y. H. Lee, S. H. Suh, Y. J. Oh, T. Y. Kim, M. H. Oh, C. J. Kim, B. K. Ju, "Uncooled thermopile infrared detector with chromium oxide absorption

- layer,” *Sensors and Actuators A*, vol. 95, pp. 24-28, 2001.
- [9] D. W. Porterfield, J. L. Hesler, R. Densing, E. R. Mueller, T.W. Crowe, and R. M. Weikle II, “Resonant metal-mesh bandpass filters for the far infrared,” *Appl. Opt.*, vol. 33, no. 25, pp. 6046-6052, Sept. 1994.
- [10] S.T. Chase and R. D. Joseph, “Resonant Array bandpass filters for the far infrared,” *Appl. Opt.*, vol. 22, no. 11 pp. 1775-1779, June 1983.
- [11] V. P. Tomaselli, D. C. Edewaard, P. Gillan, and K. D. Möller, “Far-infrared bandpass filter from cross-shaped grids,” *Appl. Opt.*, vol. 20, no. 8, pp. 1361-1366, April 1981.
- [12] M. E. MacDonald, A. Alexanian, R. A. York, Z. Popovic, and E. Grossman, “Spectral transmittance of lossy printed resonant-grid terahertz bandpass filters,” *IEEE Trans. Microwave Theory and Techniques*, vol. 48, no. 4, pp. 712-717, April 2000.
- [13] H. J. Liebe, “Modeling attenuation and phase of radio wave in air at frequencies below 1000 GHz,” *Radio Science*, vol. 16, no. 6, pp. 1183-1199, 1981.
- [14] B. L. Ulich, “Improved Correction for Millimeter-Wavelength Atmospheric Attenuation”, *Astrophysical Letters*, vol. 21., pp. 21-28, 1980.
- [15] M. A. Holdaway and J. R. Pardo, “Modeling of the submillimeter Opacity on Chajnantor”, NRAO MMA Memo 187, USA, 1997.
- [16] E. Grossman, “AT, atmospheric transmission software user’s manual”, Airhead Software Cp., Boulder, CO, USA, 1989.
- [17] R. Winston and W. T. Welford, “Design of nonimaging concentrators as second stages in tandem with image-forming first-stage concentrators,” *Appl. Opt.*, vol. 19, pp. 347-351, Feb.1980.
- [18] E. W. Weisstein, “Millimeter/submillimeter Fourier transform spectroscopy of jovian planet atmospheres,” Doctoral Thesis, California Institute of Technology, Pasadena, California, 1996.
- [19] P. Kaufmann, J. E. R. Costa, C. G. de Castro, Y. R. Hadano, J. S. Kingsley, R. K. Kingsley, H. Levato, A. Marun, J. P. Raulin, M. Rovira, E. Correia, V. R. Silva, “The New Submillimeter-wave Solar Telescope”, *Proc. SMBO/IEEE MTT-S International Microwave and Optoelectronics Conference*, August 6-10, Belém, PA, Brasil, 439, 2001.
- [20] A. M. Melo; C. G. Giménez de Castro; P. Kaufmann; H. Levato; A. Marun; P. Pereyra and J. P. Raulin, “Determination of submillimeter atmospheric opacity at El Leoncito, Argentina Andes”, *Telecomunicações INATEL*, vol. 6, pp. 32-36, 2003.
- [21] P. Kaufmann; C. G. Giménez de Castro; J. P. Raulin, A. M. Melo; H. Levato; A. Marun; J. L. Giuliani and P. Pereyra, “Comparison of Radiometric Methods for Measuring Atmospheric Transmission at Submillimeter Wavelengths”, in preparation, 2003.