

Artigo: Network-Coded Secondary Communication with Opportunistic Energy Harvesting

## Resumo

O artigo em questão apresenta a proposta de um framework de rádio cognitivo para comunicação cooperativa com codificação de rede e transferência sem fio de energia, chamado CEGNC. Nesse esquema, os usuários secundários (SUs) são alimentados somente através das transmissões do usuário primário (PU) e cooperam por meio do modelo EGNC, um protocolo que trabalha com comutação no tempo.

O trabalho simulou a probabilidade de falha na comunicação (outage), assim como o tempo ótimo de transferência de energia que minimiza tal valor, além de apresentar os resultados analíticos correspondentes. Ele ainda comparou os resultados com os esquemas CEDT, que realiza transmissão direta (não cooperativa) e CEDF, que decodifica e encaminha a mensagem. Finalmente, foi concluído que o esquema CEGNC pode aumentar consideravelmente o desempenho do secundário.

## Metodologia

O estudo foi feito em um cenário de interferência limitada, em que os SUs estão posicionados muito próximos do PU. Assim, embora a interferência do primário prejudique o desempenho dos secundários, ela consiste em uma fonte de energia para os SUs. O modelo do sistema em questão é apresentado na Figura 1.

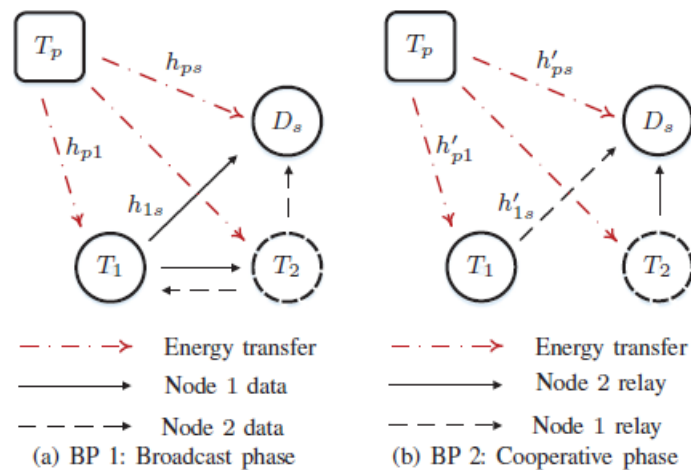


Figura 1 – Modelo do sistema

Nesse cenário, a rede secundária, composta por  $T_1$ ,  $T_2$  e  $D_s$ , é alimentada somente por meio da transferência de energia de  $T_p$ , que integra a rede primária. E  $T_1$  e  $T_2$  vão transmitir dados para  $D_s$  em um esquema de cooperação. Assim, cada slot de tempo é dividido em três partes:

a primeira, com duração  $\alpha$ , é reservada para a transmissão de energia de  $T_p$ , a segunda, com duração  $k_1$ , consiste na fase de broadcast, em que  $T_1$  e  $T_2$  transmitem seus dados a  $D_s$ , e a terceira, com duração  $k_2$ , é dedicada à cooperação dos nós secundários que retransmitem a informação um do outro.

A probabilidade de falha na comunicação (outage) é tida como a probabilidade de a informação mútua entre dois nós cair abaixo de uma determinada taxa  $R$ . Assim, considerando o modelo CEDT:

$$Out_{ET} = \Pr \left\{ \log_2 \left( 1 + \frac{2\eta\alpha|h_{ij}|^2|h_{pi}|^2}{(1-\alpha)|h_{pj}|^2} \right) < \frac{R}{(1-\alpha)} \right\}$$

Já para os modelos CEGNC e CEDF, pode-se calcular a probabilidade de outage a partir da expressão anterior considerando-se o ganho de código e a ordem de diversidade de cada modelo. Assim, o valor é calculado, respectivamente por:

$$Out_{CEGNC} = C_{(k_1+k_2-1), k_2} Out_{ET}^{2+k_2}$$

$$Out_{CEDF} = \frac{1}{2} Out_{ET}^2$$

## Simulação

Esta simulação vai apresentar os valores da probabilidade de outage para toda a faixa de valores de  $\alpha$  ( $0 < \alpha < 1$ ). Os demais parâmetros são:  $\eta = 1$ ,  $R = 2$ ,  $k_1 = k_2 = 2$ , e as variáveis de desvanecimento de canal  $h$  possuem natureza aleatória com distribuição exponencial. As curvas simuladas são mostradas na Figura 2. Em seguida, está o código em Python utilizado.

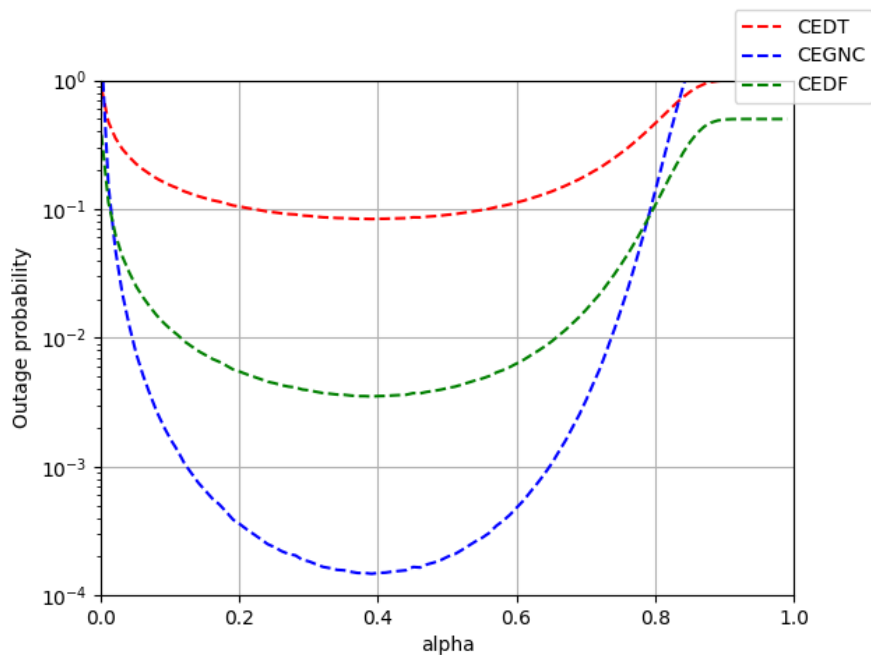


Figura 2 – Probabilidade de outage x  $\alpha$  para os modelos CEDT, CEGNC e CEDF

```

import numpy as np
import matplotlib.pyplot as plt

N = 1000000 # numero de amostras

lambda_ij = 256
lambda_pi = 1
lambda_pj = 1

k1 = k2 = 2
eta = 1
R = 2

code_gain_cegnc = 3
code_gain_cedf = 0.5
div_order_cegnc = 4
div_order_cedf = 2

OP_cedt = []
OP_cegnc = []
OP_cedf = []

for a in np.arange(0.001, 0.999, 0.01): # Valores de alpha

    h_ij = np.random.exponential(lambda_ij, N) #  $|h_{ij}|^2$ 
    h_pi = np.random.exponential(lambda_pi, N) #  $|h_{pi}|^2$ 
    h_pj = np.random.exponential(lambda_pj, N) #  $|h_{pj}|^2$ 

    I_ij = np.log2(1 + 2 * eta * a * h_ij * h_pi / (h_pj * (1 - a)))
# informação mútua

    Prob_cedt = np.sum(I_ij < (R / (1 - a))) / N
    Prob_cegnc = code_gain_cegnc * (Prob_cedt ** div_order_cegnc)
    Prob_cedf = code_gain_cedf * (Prob_cedt ** div_order_cedf)

    OP_cedt = np.append(OP_cedt, Prob_cedt)
    OP_cegnc = np.append(OP_cegnc, Prob_cegnc)
    OP_cedf = np.append(OP_cedf, Prob_cedf)

eixo_alpha = np.arange(0.001, 0.999, 0.01)

plt.semilogy(eixo_alpha, OP_cedt, 'r--')

```

```
plt.semilogy(eixo_alpha, OP_cegnc, 'b--')
plt.semilogy(eixo_alpha, OP_cedf, 'g--')

plt.grid()
plt.axis([0, 1, 1e-4, 1])
plt.xlabel('alpha')
plt.ylabel('Outage probability')
plt.figlegend(['CEDT', 'CEGNC', 'CEDF'])

plt.show()
```