

# LMT-MAC - Um Protocolo MAC Multicanal Livre de Colisões para Redes de Sensores Sem Fio

Gilson Miranda Júnior  
Ariel Felipe Ferreira Marques  
Luiz Henrique Andrade Correia

Programa de Pós-Graduação em Ciência da Computação  
Universidade Federal de Lavras

01 de Junho de 2016



GRUBi - Com

- 1 **Introdução**
- 2 **Objetivo**
- 3 **Trabalhos Relacionados**
- 4 **Lightweight Multichannel Tree MAC**
- 5 **Metodologia**
- 6 **Resultados e Discussão**
  - Taxa de Entrega
  - Latência
  - Taxa de Transmissão
- 7 **Conclusões e Trabalhos Futuros**



1 **Introdução**

2 **Objetivo**

3 **Trabalhos Relacionados**  
4 **Lightweight Multichannel  
Tree MAC**

5 **Metodologia**

6 **Resultados e Discussão**

- Taxa de Entrega
- Latência
- Taxa de Transmissão

7 **Conclusões e Trabalhos  
Futuros**



- Primeiras aplicações propostas para RSSF: monitoramento ambiental, aplicações tolerantes a falhas e atrasos [Romer and Mattern 2004].
- Desenvolvimento de dispositivos e protocolos mais robustos. Propostas de RSSF para aplicações industriais e de tempo real [Gungor and Hancke 2009].
- Operação em único canal - decisões de projeto e restrições de hardware.
- Coordenação das comunicações em protocolos multicanais (e TDMA).



1 Introdução

**2 Objetivo**

3 Trabalhos Relacionados

4 Lightweight Multichannel  
Tree MAC

5 Metodologia

6 **Resultados e Discussão**

- Taxa de Entrega
- Latência
- Taxa de Transmissão

7 **Conclusões e Trabalhos  
Futuros**



Desenvolver um protocolo MAC multicanal que permita **previsibilidade** da latência dos nós, alta vazão e alta taxa de entrega, além de demandar poucos recursos de hardware.



- 1 Introdução
- 2 Objetivo
- 3 Trabalhos Relacionados**
- 4 Lightweight Multichannel Tree MAC
- 5 Metodologia

- 6 **Resultados e Discussão**
  - Taxa de Entrega
  - Latência
  - Taxa de Transmissão
- 7 **Conclusões e Trabalhos Futuros**



- RTH-MAC - Escalonamento centralizado.
- PriorityMAC - Modo de acesso diferente de acordo com prioridade dos pacotes.
- MC-LMAC - Escalonamento dos receptores.
- TreeMAC - Distribuição justa de fatias de tempo.





- 1 Introdução
- 2 Objetivo
- 3 Trabalhos Relacionados
- 4 Lightweight Multichannel Tree MAC**
- 5 Metodologia

- 6 **Resultados e Discussão**
  - Taxa de Entrega
  - Latência
  - Taxa de Transmissão
- 7 **Conclusões e Trabalhos Futuros**

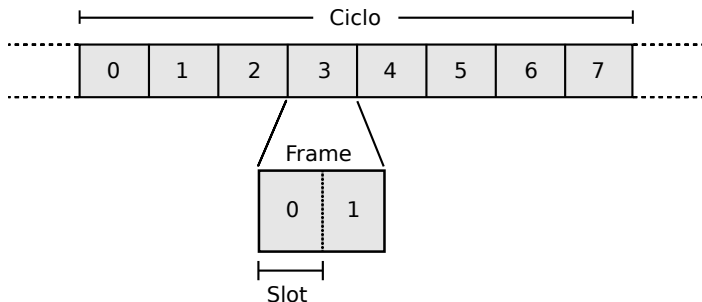


- *Lightweight Multichannel Tree MAC* - Protocolo que visa ser leve, para que possa ser implementado em nós com recursos limitados, ao mesmo tempo em que oferece latência previsível, bem como altas taxas de vazão e entrega.
- Usa como base a organização em árvore e conceitos de *frames* e *slots* semelhantes ao protocolo TreeMAC [Song et al. 2009].
- Implementado em Arduino Uno com rádio nRF24l01+ e posteriormente no simulador OMNet++.

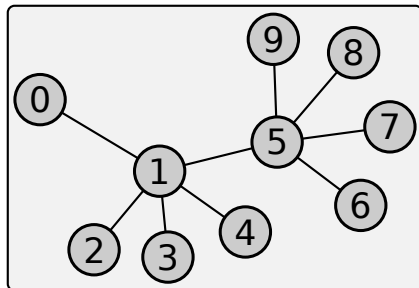


# Divisão de Tempo e Sincronização

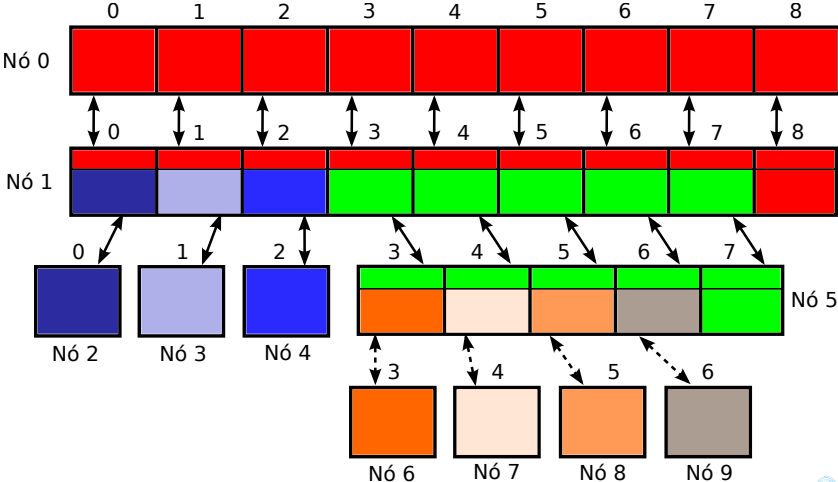
- Sincronização global de tempo.
- Divisão de um ciclo em vários *frames*.
- Cada *frame* é dividido em dois *slots*.



# Distribuição de Frames



# Distribuição de Frames



# Definição de Slots

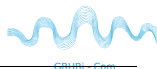
- Os nós definem seu *slot* de transmissão dentro de cada *frame* de acordo com sua profundidade.
- Em seguida, os nós definem em qual canal deve configurar o rádio em cada *slot*.

$$SlotTX \in \{0, 1\} = (profundidade - 1) \bmod 2 \quad (1)$$



# Definição de Canais

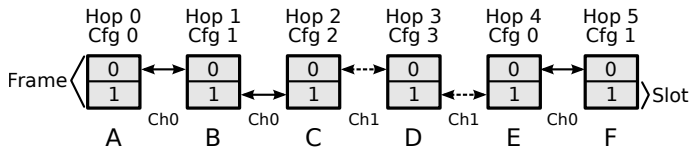
```
1: channel_count = NUM_OF_CHANNELS;
2: configurations = channel_count * 2;
3: ch_slot_0 = 0;
4: ch_slot_1 = channel_count - 1;
5: for (i = 0; i < configurations; i++) do
6:     config[i].slot_0 = ch_slot_0;
7:     config[i].slot_1 = ch_slot_1;
8:     if (i mod 2) then
9:         ch_slot_0++;
10:        ch_slot_0 = ch_slot_0 mod channel_count;
11:     else
12:         ch_slot_1++;
13:        ch_slot_1 = ch_slot_1 mod channel_count;
14:     end if
15: end for
16: slot0.channel = config[depth mod configurations].slot_0;
17: slot1.channel = config[depth mod configurations].slot_1;
```



# Configurações Slot/Canal

**Tabela:** Configurações para uma rede com dois canais.

$depth \bmod 4$	Canal no <i>slot</i> 0	Canal no <i>slot</i> 1
0	0	1
1	0	0
2	1	0
3	1	1





A latência máxima entre um nó  $u$  e o *sink* pode ser estimada pela equação:

$$L_{Max}u = T_{slot} \times (\ell_u + ((|F_{LMT-MAC}| - |F_u|) \times 2) + 1)$$

onde:

- $T_{slot}$ : tempo de duração de um *slot*.
- $\ell_u$ : profundidade do nó  $u$  na árvore (saltos para o *sink*).
- $F_{LMT-MAC}$ : conjunto de todos os *frames* da rede.
- $F_u$ : conjunto de *frames* atribuídos ao nó  $u$ .



- 1 Introdução
- 2 Objetivo
- 3 Trabalhos Relacionados
- 4 Lightweight Multichannel Tree MAC
- 5 Metodologia**

- 6 **Resultados e Discussão**
  - Taxa de Entrega
  - Latência
  - Taxa de Transmissão
- 7 **Conclusões e Trabalhos Futuros**

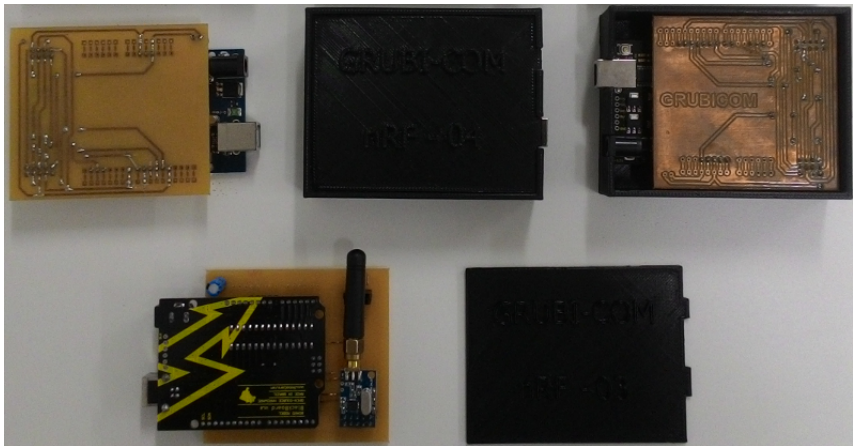


- O LMT-MAC foi comparado com o TreeMAC em simulações com 3 topologias diferentes, variando configurações de tráfego e tamanho de *buffer*.
- Foram comparados os resultados de taxa de entrega, latência até o *sink* e vazão.

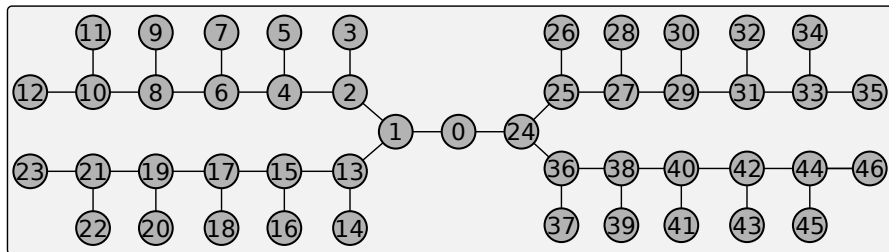
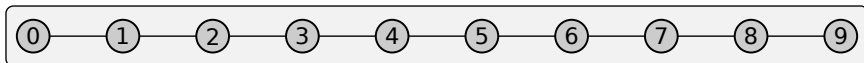


# Testes reais

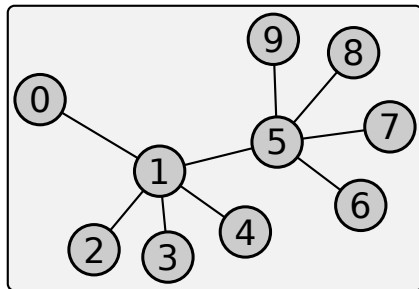
- Testes em nós reais foram executados com 10 nós baseados em Arduino Uno e rádio Nordic nRF24I01+.



# Topologias 1 (Linear) e 2 (Árvore Binária)



# Topologia 3 (Arbitrária)



# Parâmetros de Simulação

Parâmetro	Topologia		
	Linear	Árvore	Arbitrária
Número de nós	30	47	10
Número de <i>frames</i>	29	46	9
Tamanho de <i>slot</i>	20 ms		
Tempo de guarda	1 ms		
Taxa de transmissão	2 Mbps		
Potência de transmissão	0 dBm (1 mW)		
Tráfego (pps)	1, 5, 20, 40		
Tamanho de <i>buffer</i> (pacotes)	1, 5, 10, 20		
Tempo de simulação	2000 s		



# Parâmetros dos Testes Reais

Parâmetro	Valor
Número de nós	10
Número de <i>frames</i>	10
Tamanho de <i>slot</i>	100 ms
Tempo de guarda	20 ms
Taxa de transmissão	2 Mbps
Potência de transmissão	0 dBm (1 mW)
Tamanho de <i>payload</i>	32 Bytes
Capacidade de <i>buffer</i>	20 pacotes



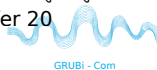
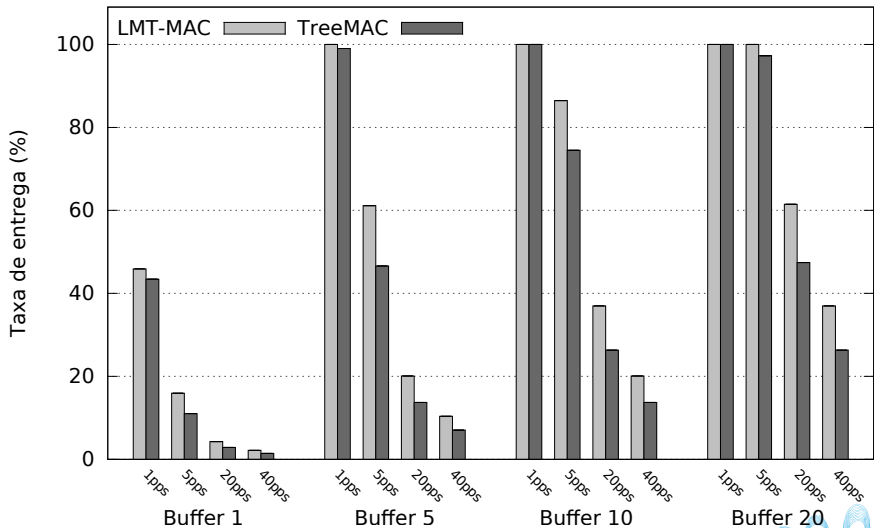


- 1 Introdução
- 2 Objetivo
- 3 Trabalhos Relacionados
- 4 Lightweight Multichannel Tree MAC
- 5 Metodologia

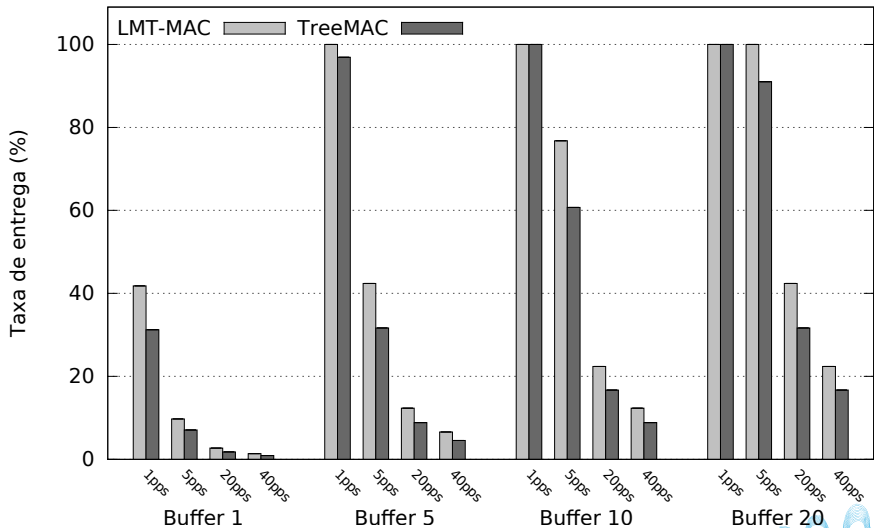
- 6 **Resultados e Discussão**
  - Taxa de Entrega
  - Latência
  - Taxa de Transmissão
- 7 Conclusões e Trabalhos Futuros



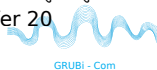
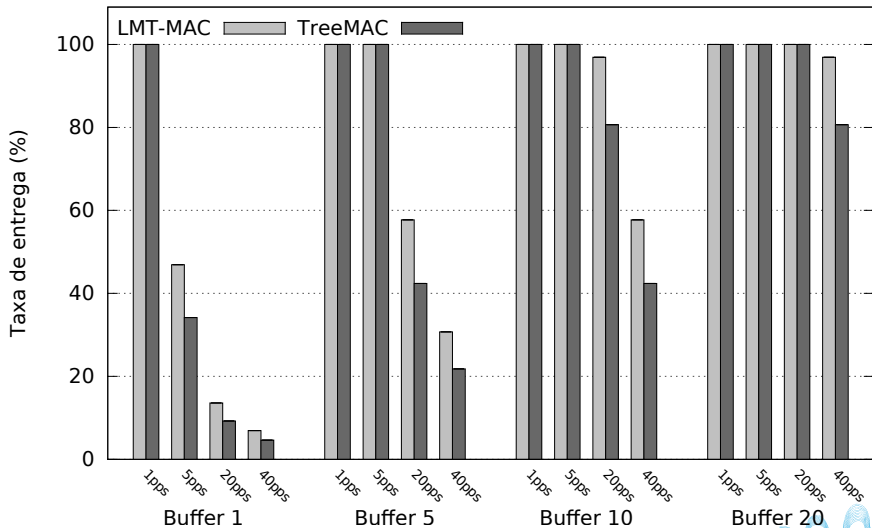
# Taxa de Entrega - Topologia Linear



# Taxa de Entrega - Árvore Binária

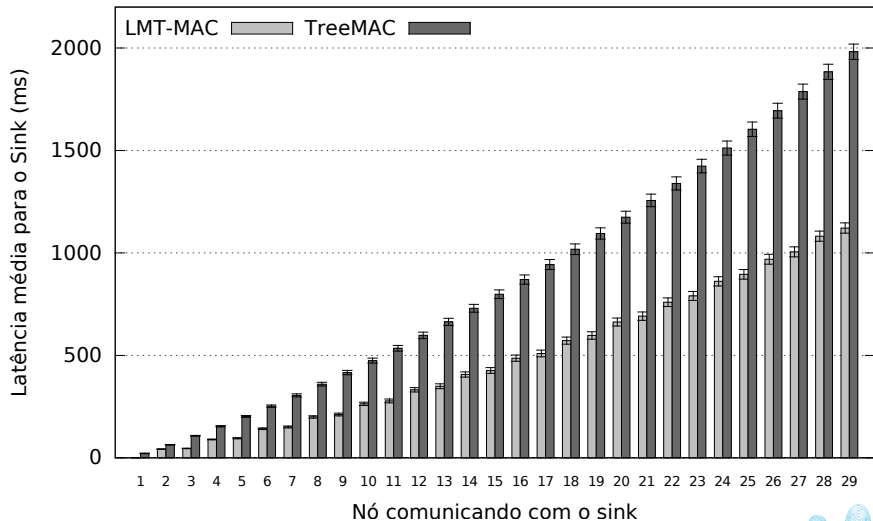


# Taxa de Entrega - Topologia Arbitrária

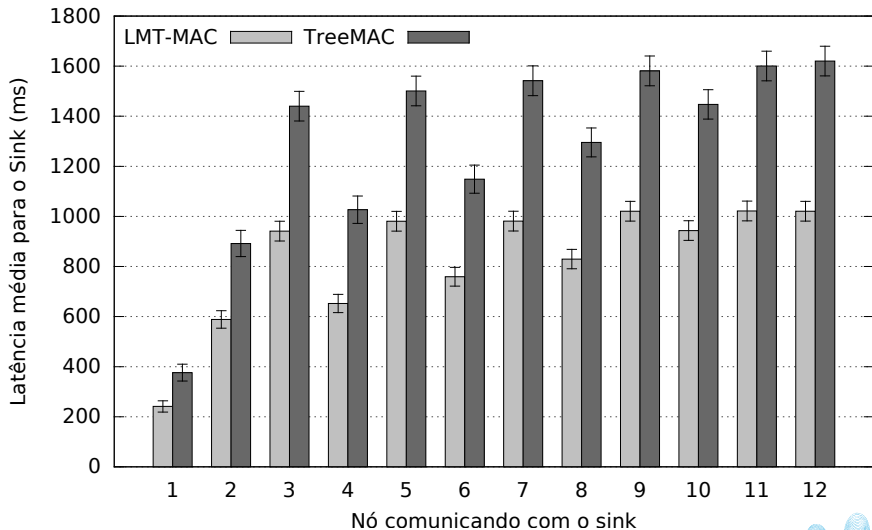


GRUBi - Com

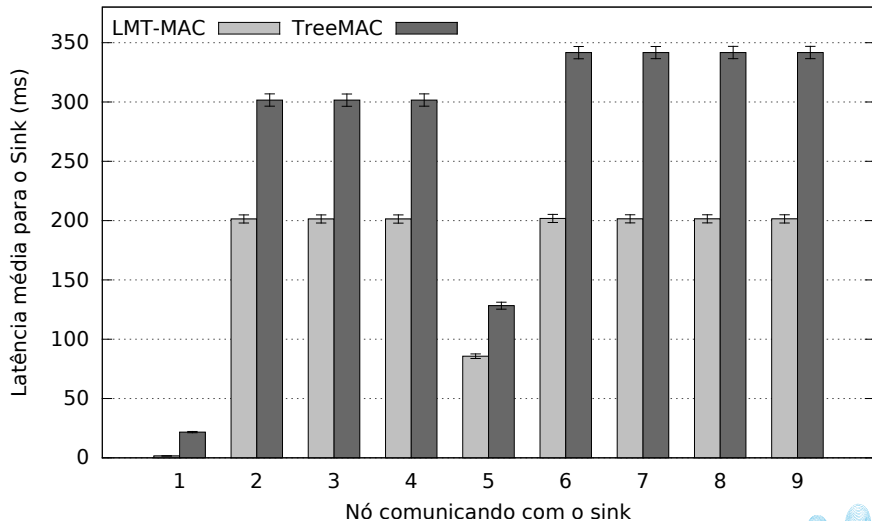
# Latência - Topologia Linear



# Latência - Árvore Binária



# Latência - Topologia Arbitrária

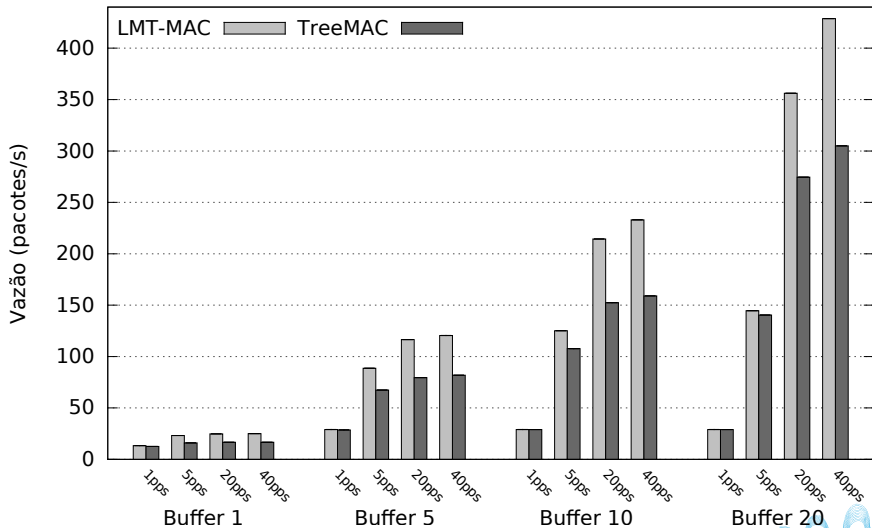


# Resultados - Latência

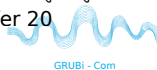
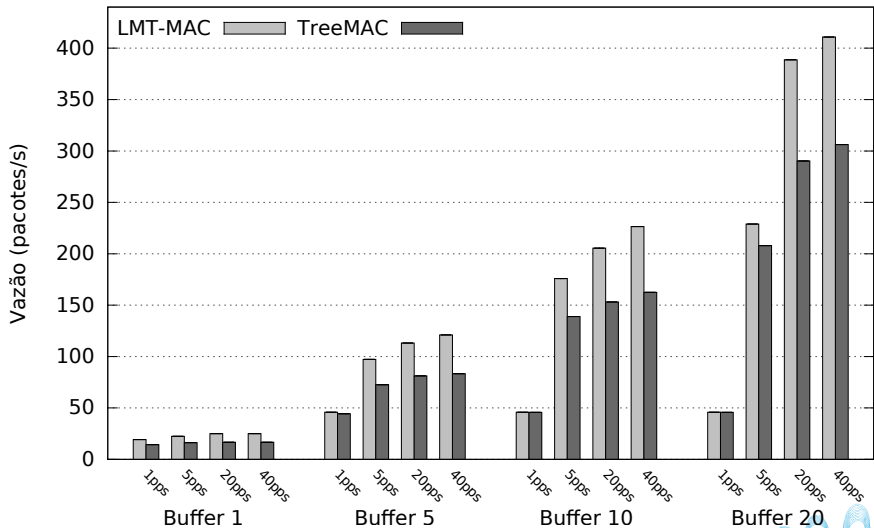
Origem → Destino	Salto	Tempo mínimo	Tempo máximo	Tempo médio
Sink → Nó 1	1	31,75 ms	398,66 ms	141,108 ms
Sink → Nó 2	2	201,83 ms	2.292,98 ms	1.248,86 ms
Sink → Nó 5	2	162,26 ms	1.494,80 ms	656,55 ms
Sink → Nó 6	3	1.435,89 ms	4.888,61 ms	3.019,09 ms
Nó 1 → Sink	1	1,66 ms	397,59 ms	115,33 ms
Nó 2 → Sink	2	115,50 ms	4.097,30 ms	2.091,11 ms
Nó 5 → Sink	2	103,54 ms	2.495,06 ms	920,86 ms
Nó 6 → Sink	3	312,90 ms	8.221,24 ms	4.362,53 ms



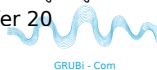
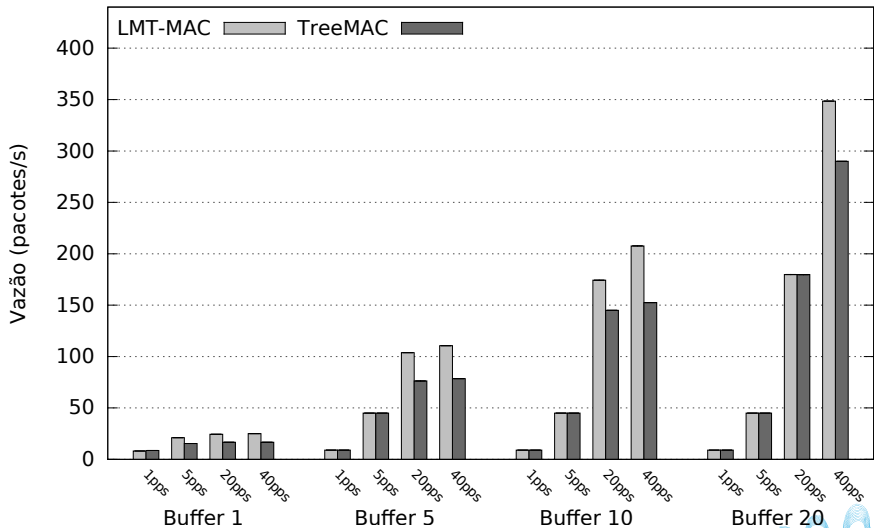
# Taxa de Transmissão - Topologia Linear



# Taxa de Transmissão - Árvore Binária



# Taxa de Transmissão - Topologia Arbitrária



# Taxa de Transmissão Ponto-a-Ponto (nós reais)

	Quadro (nRF24I01+)		Pacote do protocolo		Quadro Resultante	
Cabeçalho	65 bits	20,25%	88 bits	34,37%	153 bits	47,66%
Dados	256 bits	79,75%	168 bits	65,63%	168 bits	52,33%
Total	321 bits	100%	256 bits	100%	321 bits	100%

	Nó filho → Nó pai	Nó pai → Nó filho
Pacotes enviados/frame	20	20
Tempo médio de transmissão	23,72ms ( $\pm 0,043ms$ )	23,65ms ( $\pm 0,005ms$ )
Velocidade	270.608bps ( $\pm 490bps$ )	271.411bps ( $\pm 59bps$ )
Velocidade de transmissão de dados da aplicação	141.544bps ( $\pm 256bps$ )	142.047bps ( $\pm 31bps$ )
Total transmitido/frame	802,5B	802,5B
Total de dados/frame	420B (52,33%)	420B (52,33%)
Overhead	382,5B (47,66%)	382,5B (47,66%)



- 1 Introdução
- 2 Objetivo
- 3 Trabalhos Relacionados
- 4 Lightweight Multichannel Tree MAC
- 5 Metodologia

- 6 Resultados e Discussão
  - Taxa de Entrega
  - Latência
  - Taxa de Transmissão
- 7 **Conclusões e Trabalhos Futuros**



## Conclusões

- O LMT-MAC é capaz de prover comunicação multicanal sem o uso de algoritmos complexos para a alocação de canais.
- A latência obtida nas simulações foi coerente com a prevista.
- O hardware de baixo custo e de montagem simples permite a sua utilização em futuras pesquisas na área de RSSF.

## Trabalhos Futuros

- Integrar um mecanismo de uso adaptativo de *slots*.
- Adicionar suporte a mobilidade.
- Melhorias no protocolo de sincronização da implementação real.



Obrigado!

Gilson Miranda Júnior - junior.kdm@gmail.com

Ariel Felipe Ferreira Marques - arielffmarques@gmail.com

Luiz Henrique Andrade Correia - lcorreia@dcc.ufla.br





Gungor, V. and Hancke, G. (2009).

Industrial wireless sensor networks: Challenges, design principles, and technical approaches.  
*Industrial Electronics, IEEE Transactions on*, 56(10):4258–4265.



Romer, K. and Mattern, F. (2004).

The design space of wireless sensor networks.  
*Wireless Communications, IEEE*, 11(6):54–61.



Song, W.-Z., Huang, R., Shirazi, B., and LaHusen, R. (2009).

TreeMAC: Localized TDMA MAC protocol for real-time high-data-rate sensor networks.  
*Pervasive and Mobile Computing*, 5(6):750 – 765.  
PerCom 2009.

