

Transmissão Multicast Confiável e Experimentos na Internet

Jorge Allyson Azevedo

Milena Scanferla, Daniel Sadoc Menasché
Edmundo A. de Souza e Silva

LAND - UFRJ

Maio 2003

Roteiro

1 Introdução

2 RML: uma biblioteca para transmissão multicast confiável

3 Tangram Whiteboard

4 Resultados dos Experimentos e o modelo desenvolvido

5 Conclusão

1

2

3

4

5

Introdução

- ➡ Crescimento e popularização das redes de computadores abrem espaço para novas aplicações
- ➡ Entre essas novas aplicações se destacam as ferramentas para trabalho cooperativo e ensino a distância
- ➡ Características principais:
 - ➡ Envio de um mesmo conjunto de dados para um determinado grupo de usuários
 - ➡ Diferentes níveis de interatividade

Introdução (cont.)

1

➔ Exemplos de ferramentas desenvolvidas no LAND:

2

➔ VivaVoz: aplicativo para transmissão de voz sobre IP

4

➔ Servidor Multimídia: armazena arquivos multimídia como aulas gravadas em vídeo

5

➔ **Whiteboard**: interação entre usuários através de figuras e pequenos textos em uma área para desenho

➔ Necessidade de um *whiteboard* para:

➔ desenvolvimento cooperativo de modelos no TANGRAM-II

➔ reuniões e aulas virtuais

1

2

3

4

5

➡ Necessidade de melhor aproveitamento dos recursos de rede

➡ Utilização de Transmissão Multicast

➡ Vantagens Transmissão Multicast:

➡ Somente uma cópia dos dados é enviada pela rede

➡ Não é necessário conhecer previamente os membros da sessão

1

2

3

4

5

➡ Desvantagens da Transmissão Multicast

- ➡ Não oferece confiabilidade - os dados podem ser perdidos e chegar fora de ordem
- ➡ Atualmente nem todos os roteadores tem o multicast habilitado e configurado

➡ Objetivos deste trabalho:

- ➡ Implementar uma biblioteca de funções para transmissão multicast confiável
- ➡ Desenvolver a ferramenta Tangram Whiteboard

Transmissão Multicast Confiável

1

2

3

4

5

- ➔ O IP Multicast não oferece confiabilidade
- ➔ Implementação dos mecanismos para garantir a confiabilidade pode ser feita:
 - ➔ No nível de rede
 - ➔ No nível da aplicação
- ➔ A implementação no nível da aplicação mantém a simplicidade da rede, característica que facilitou o crescimento da Internet

Transmissão Multicast Confiável

1

2

3

4

5

➡ Propostas para transmissão multicast confiável se dividem em 3 grandes grupos:

➡ **Grupo 1** - as propostas que sugerem a implementação de novas rotinas nos roteadores. Ex: *Active Reliable Multicast*

➡ **Grupo 2** - as propostas que sugerem a implementação de novas rotinas na aplicação e utilização do IP Multicast para transmissão dos dados. Ex: *Scalable Reliable Multicast*

➡ **Grupo 3** - as propostas que sugerem a utilização de IP Unicast para transmissão dos dados. Ex: *Narada*

Transmissão Multicast Confiável

1

2

3

4

5



EMISSOR



DADOS 1

RECEPTORES



➡ Protocolos baseados nos receptores: RINA
Passo 1

Transmissão Multicast Confiável

1

2

3

4

5



EMISSOR



DADOS 2

RECEPTORES



OK



➡ Passo 2

Transmissão Multicast Confiável

1

2

3

4

5



EMISSOR



RECEPTORES



OK



➡ Passo 3

Transmissão Multicast Confiável

1

2

3

4

5



EMISSOR



RECEPTORES



OK



➡ Passo 4

Transmissão Multicast Confiável

- 1
- 2
- 3
- 4
- 5



EMISSOR

RECEPTORES



OK



OK



OK

➡ Passo 5

1

2

Protocolos baseados no receptor

3

➡ Vantagens

4

➡ O emissor não precisa conhecer o grupo de receptores

5

➡ O emissor não precisa processar ACKs dos receptores

➡ Desvantagem

➡ Não existe mecanismo que indique explicitamente quando os dados podem ser liberados da memória

RML - *Reliable Multicast Library*

1

2

3

4

5

➡ Biblioteca de funções que pode ser usada por aplicações que necessitem fazer transmissão multicast confiável

➡ Garante a entrega dos dados em ordem e sem perdas

➡ Altamente configurável:

➡ Parâmetros dos temporizadores

➡ Simulação de perdas

➡ Tamanho da cache

➡ Arquivo de log

1

2

3

4

5

Temporizadores

- ➡ Utilizados para reduzir o número de mensagens duplicadas durante a recuperação das perdas de pacotes
- ➡ Escolhidos aleatoriamente nos intervalos determinados:

$$T_{nak} = (A_{nak} \cdot R, (A_{nak} + B_{nak}) \cdot R)$$

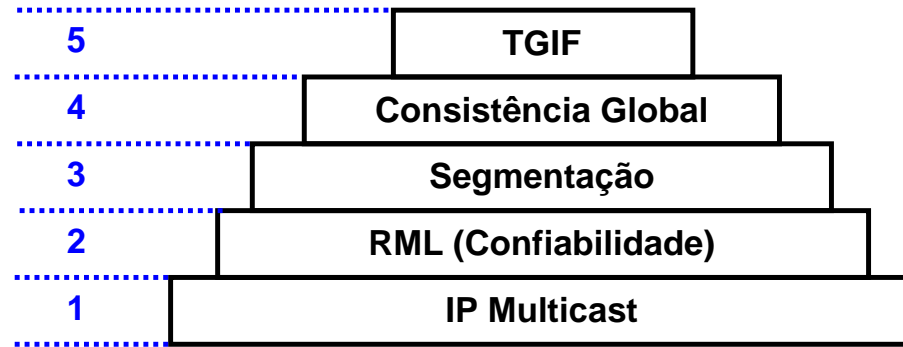
$$T_{wait} = (A_{wait} \cdot R, (A_{wait} + B_{wait}) \cdot R)$$

$$T_{ret} = (A_{ret} \cdot R, (A_{ret} + B_{ret}) \cdot R)$$

- ➡ Onde **R** representa o retardo até o emissor dos dados

1

Camadas que formam o TGWB:



2

3

4

5

- ➡ A camada 5 representa o conjunto dos comandos da ferramenta TGIF que podem ser usadas no TGWB
- ➡ A camada 4 representa as rotinas necessárias para garantir a consistência global
- ➡ A camada 3 engloba as rotinas relativa à fragmentação dos dados
- ➡ As camadas 2 e 1 contém os mecanismos para enviar, de forma confiável, os comandos da interface local para os outros membros da sessão multicast

Exemplo de Inconsistência

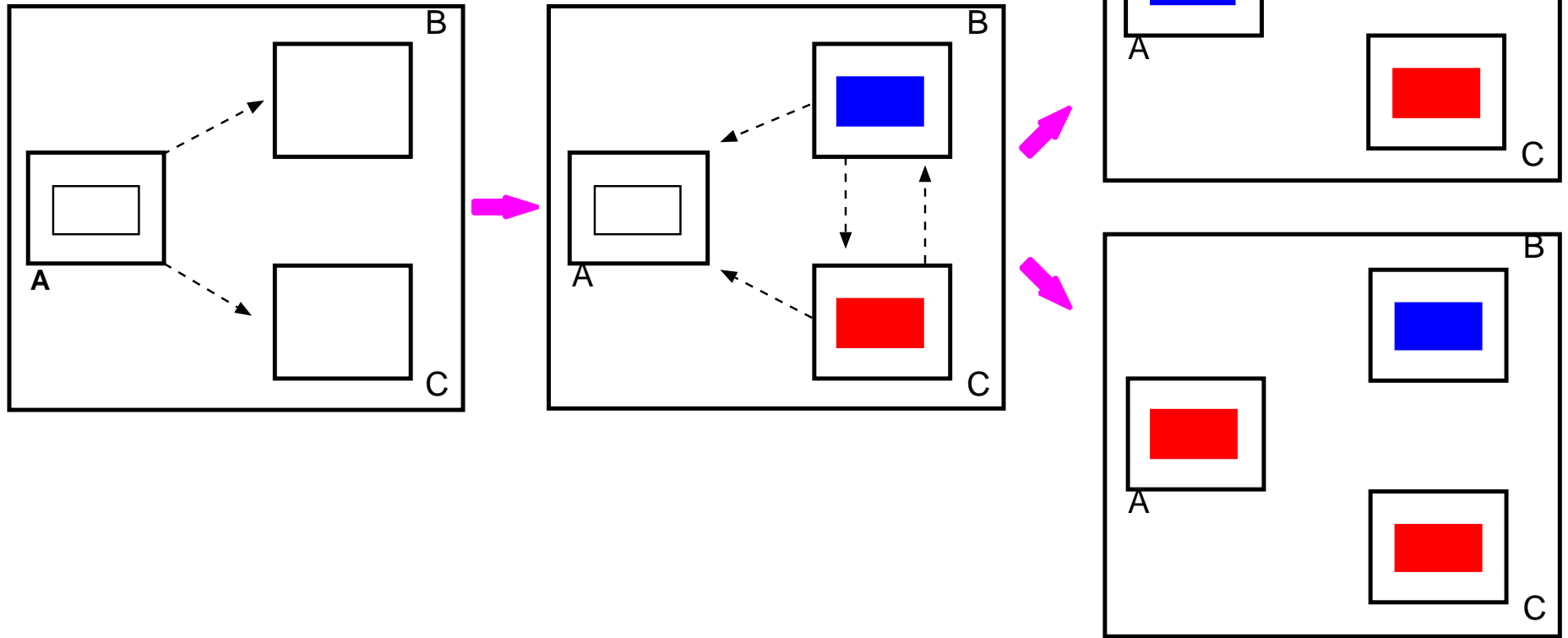
1

2

3

4

5



1 **Consistência Global**

2

➡ Definição de uma relação de ordem total para os comandos

3

4

➡ Essa relação é baseada na comparação dos valores dos:

5

➡ Relógios Lógicos

➡ Identificação dos membros da sessão

➡ O mecanismo de *Rollback and Recovery* é utilizado para refazer a ordenação dos comandos quando necessário

Tangram Whiteboard

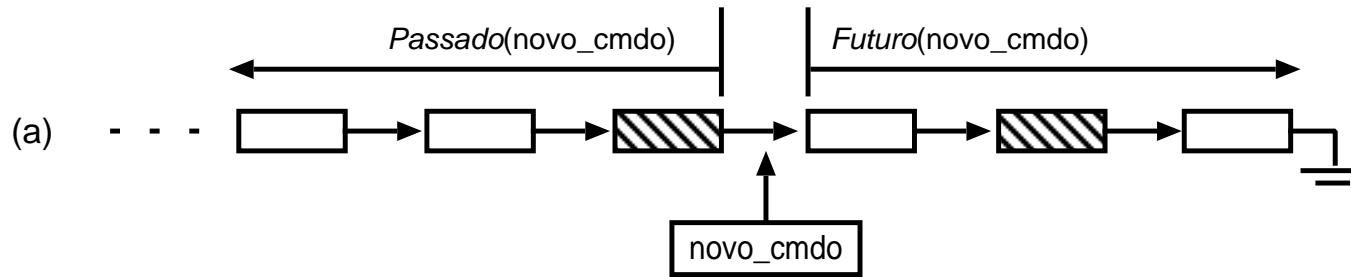
1

2

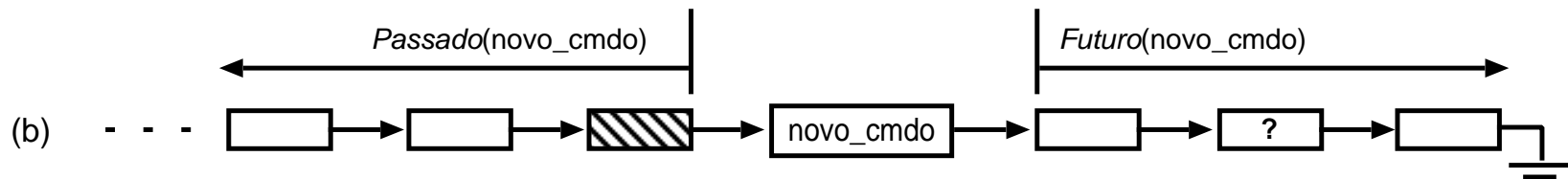
3

4

5



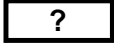


Procurar o ponto de inserção para novo_cmdo

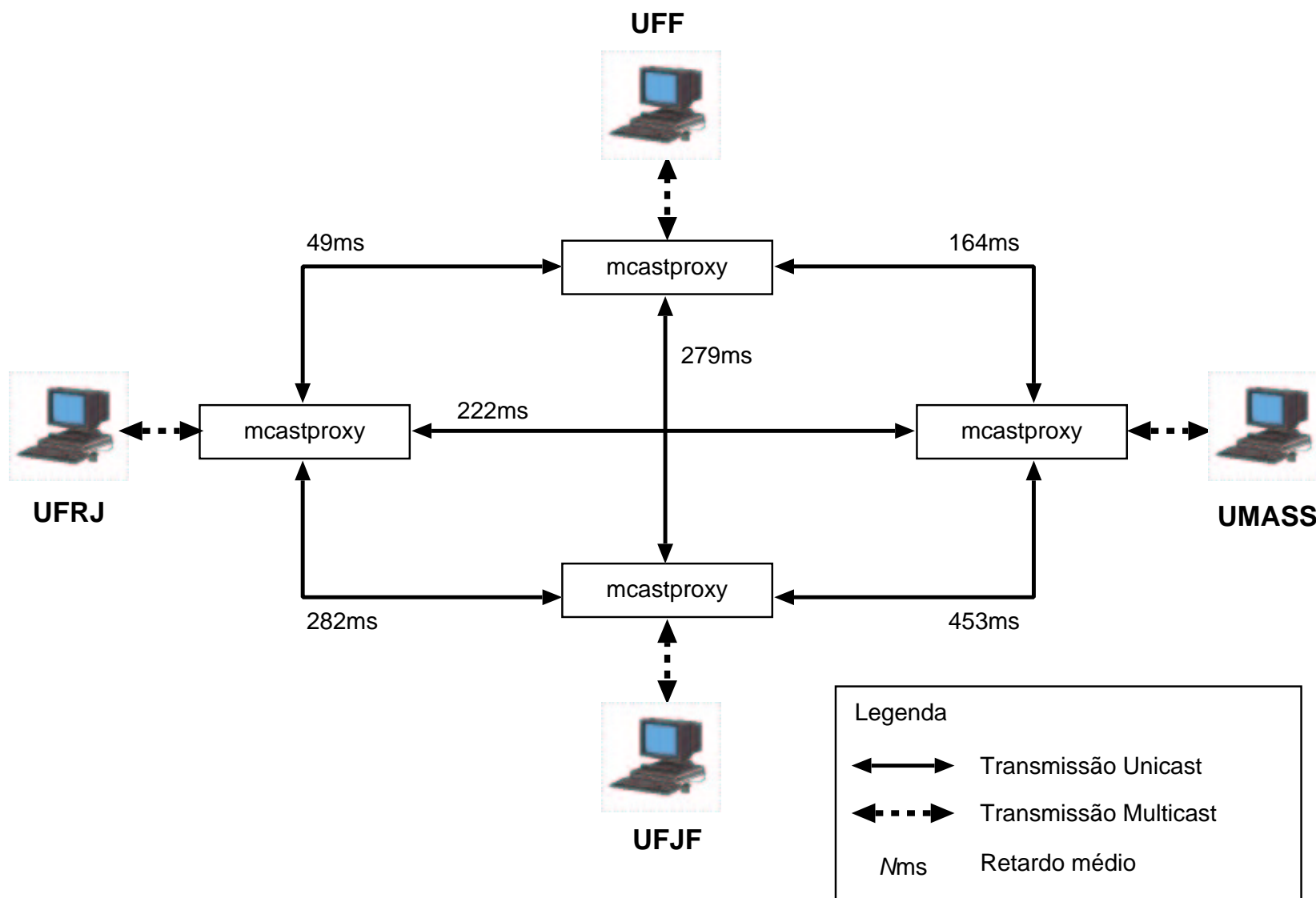


Desfazer os comandos em *Futuro(novo_cmdo)*
Inserir e executar novo_cmdo
Refazer os comandos em *Futuro(novo_cmdo)*

Estados possíveis dos comandos na lista

-  Executado
-  Inconsistente
-  Inconsistente mas que pode mudar de estado por causa da inserção de um novo comando

Resultados Experimentos/Modelo



Resultados Experimentos/Modelo

1

2

3

4

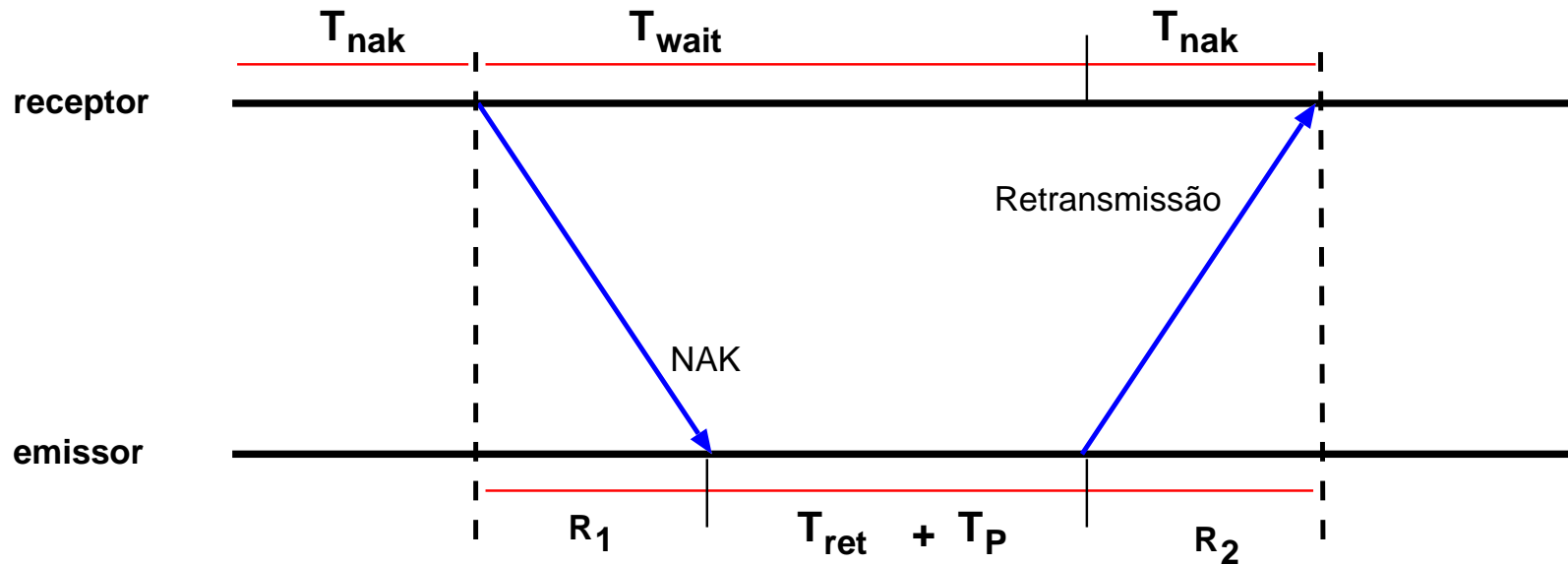
5

➡ Durante os experimentos foram representados dois cenários:

➡ **Cenário 1:** representa a utilização do TGWB em uma palestra, onde existe apenas um emissor de dados

➡ **Cenário 2:** representa a utilização do TGWB no desenvolvimento de um modelo no TANGRAM-II. Neste caso todos os membros são emissores e receptores

Relação entre os parâmetros dos temporizadores

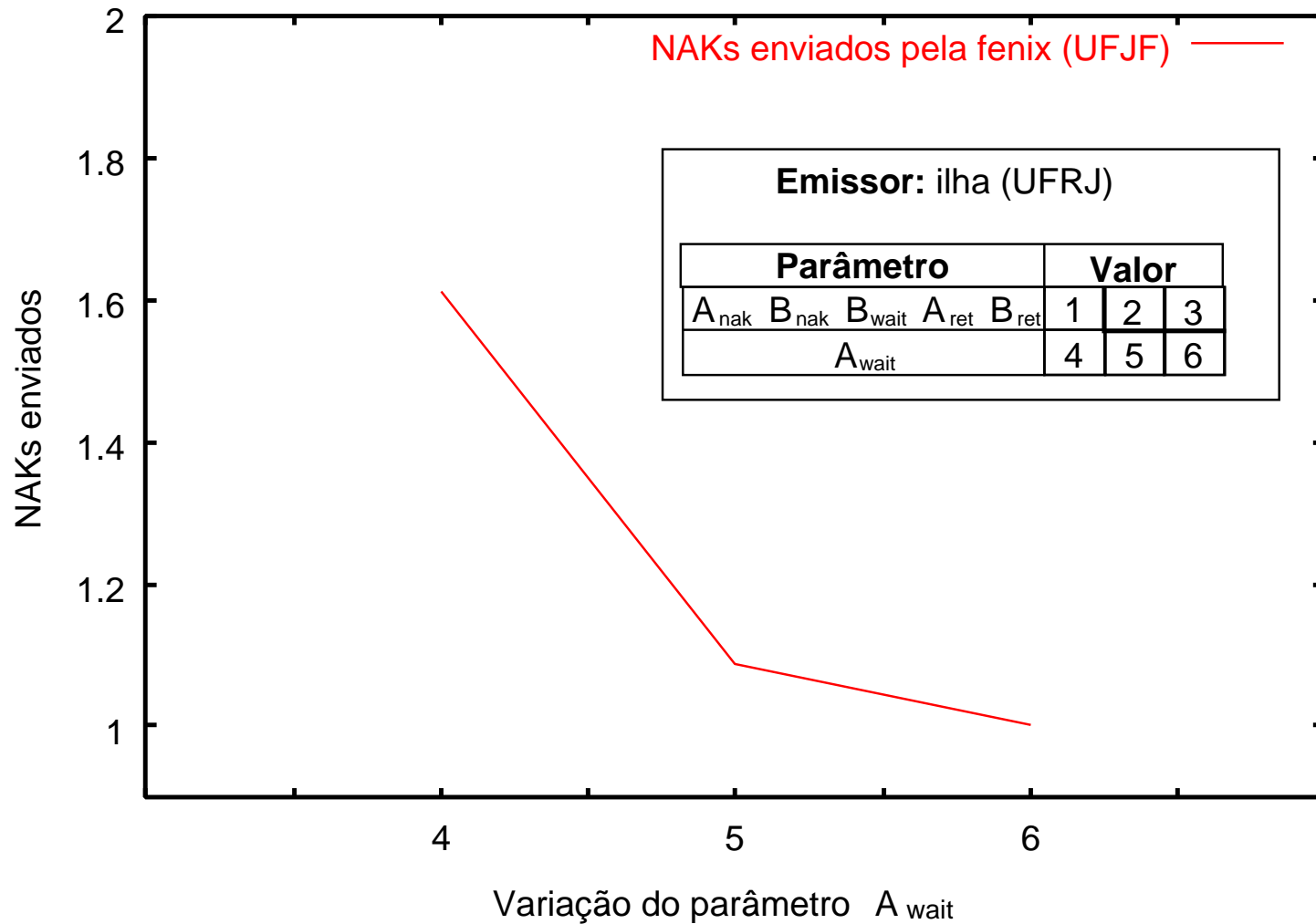


- T_{wait} temporizador para espera por retransmissão
- T_{nak} temporizador para espera para enviar NAK
- R_1, R_2 Retardos da rede
- T_{ret} temporizador para espera para enviar retransmissão
- T_p tempo de processamento de pacote

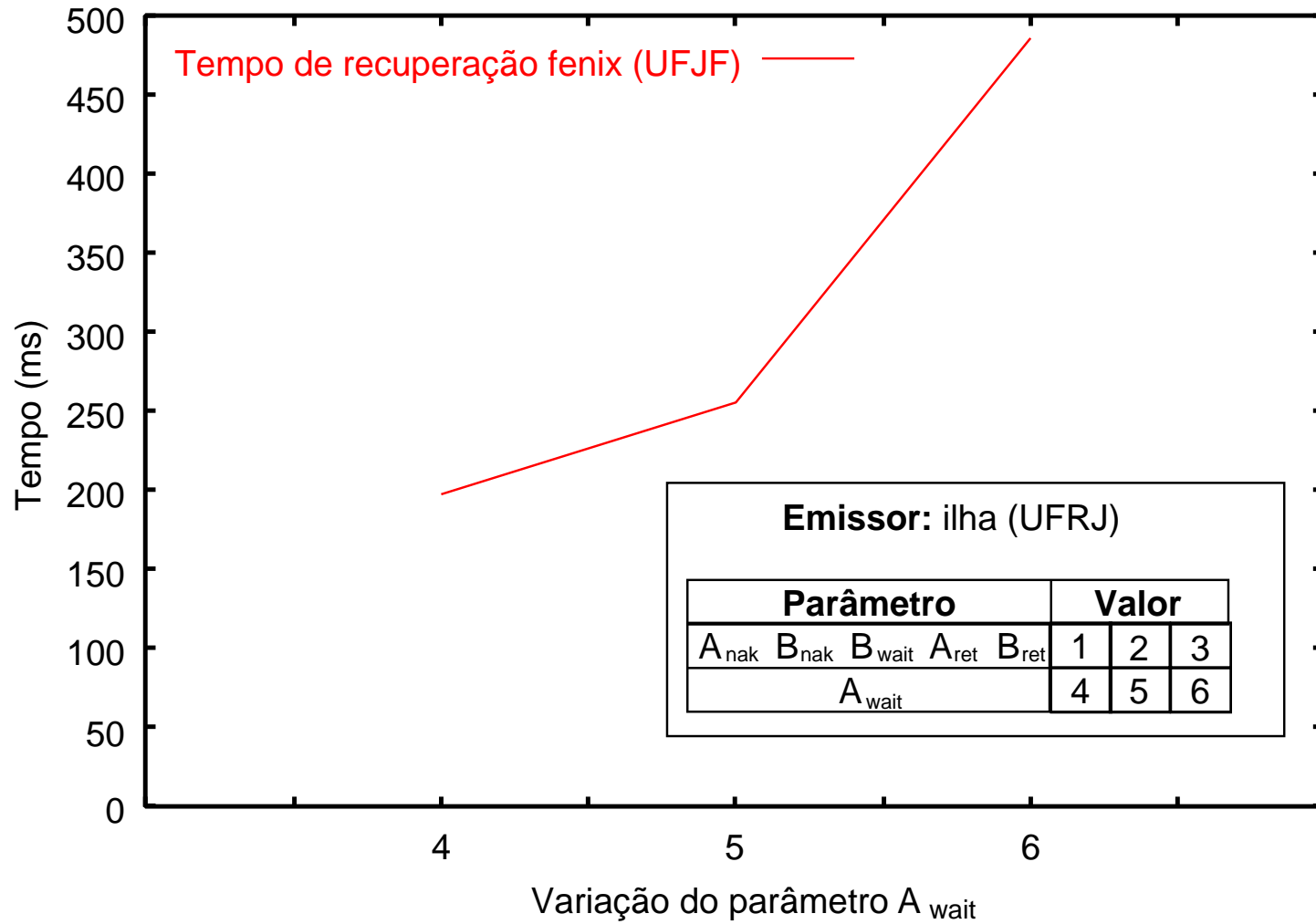
$$A_{wait} > 2 + A_{ret} + B_{ret} - A_{nak}$$

- 1
- 2
- 3
- 4
- 5

Resultados relativos ao Cenário 1

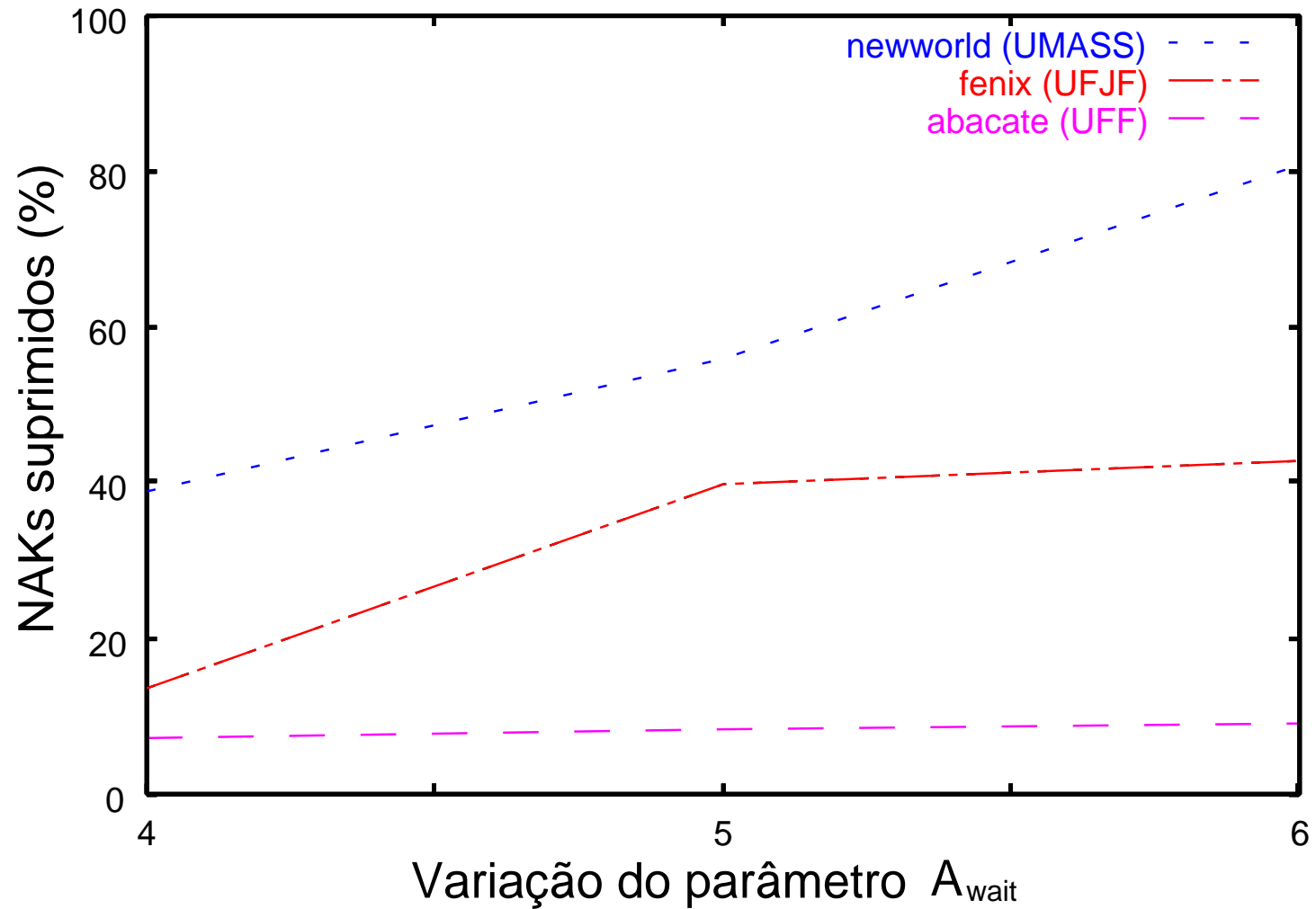


Resultados relativos ao Cenário 1



- 1
- 2
- 3
- 4
- 5

Resultados relativos ao Cenário 1



Resultados Experimentos/Modelo


1 O modelo da RML

2 Simplificações

3

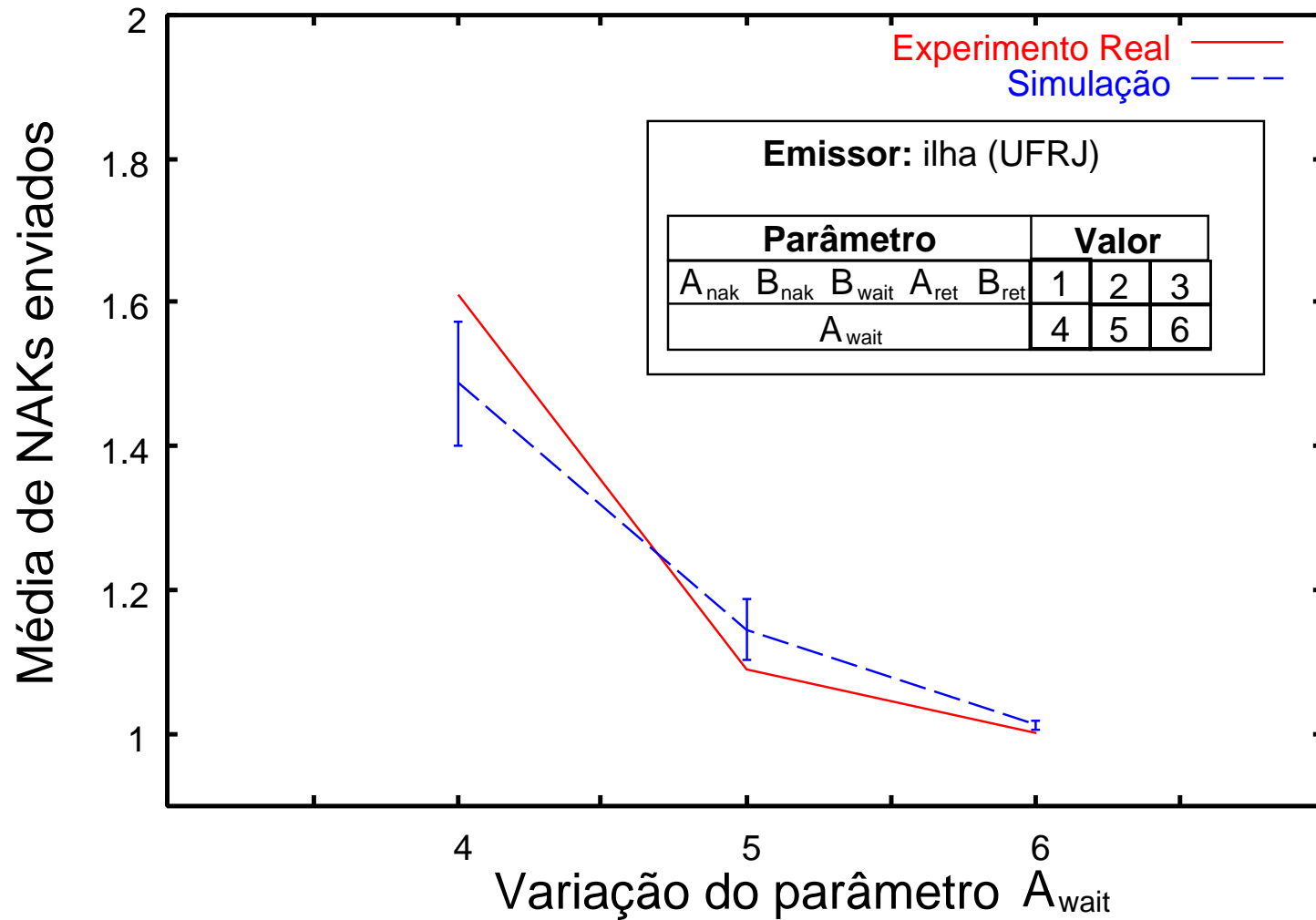
4  cache

5  Estrutura dos pacotes

 Resultados das simulações ajudaram no desenvolvimento do protocolo e identificação de erros de implementação

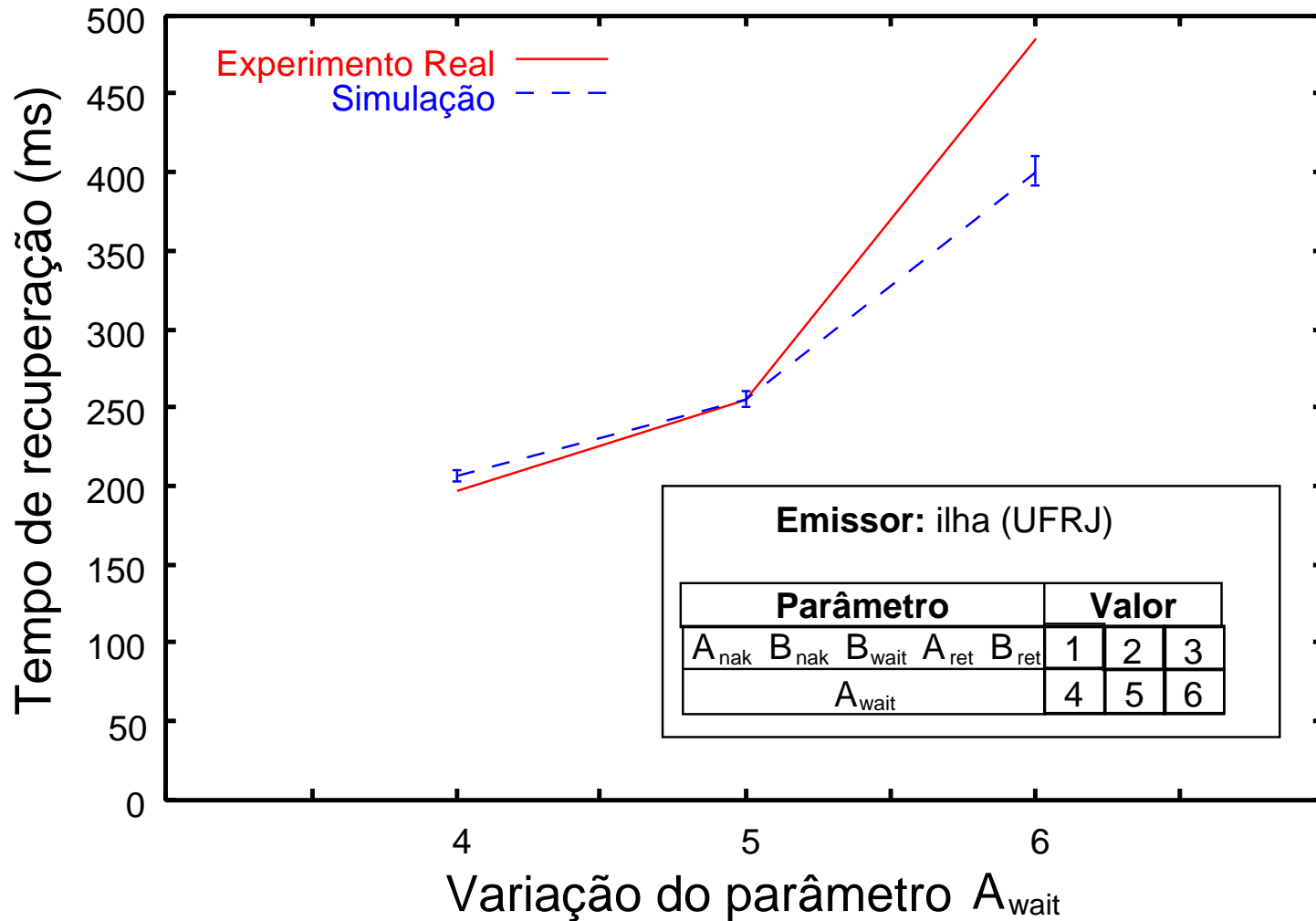
- 1
- 2
- 3
- 4
- 5

O modelo da RML



- 1
- 2
- 3
- 4
- 5

O modelo da RML



Conclusão

1

2

3

4

5

➡ Criação da biblioteca RML

➡ Desenvolvimento do Tangram Whiteboard

➡ Montagem de um ambiente de testes

➡ Resultados dos experimentos mostraram:

➡ a viabilidade da utilização da RML e do Tangram Whiteboard

➡ o impacto da variação dos parâmetros sobre o desempenho do protocolo implementado

➡ Construção do modelo do protocolo no TANGRAM-II

Trabalhos Futuros

1

2

3

4

5

➡ RML

➡ Implementação de um ajuste dinâmico para os temporizadores

➡ Tangram Whiteboard

➡ Inclusão de novas características para melhorar a interface

➡ Modelo

➡ Continuação do desenvolvimento para verificar outras características do protocolo, como a escalabilidade

➡ Desenvolvimento de multicast na aplicação

Perguntas?

Jorge Allyson Azevedo

allyson@land.ufrj.br

<http://www.land.ufrj.br/~allyson>