

View Synchronous Multicast sobre UDP

SDIS 2018-2019

Isabel Oliveira

Daniel Granhão

Beatriz Cruz





Sumário

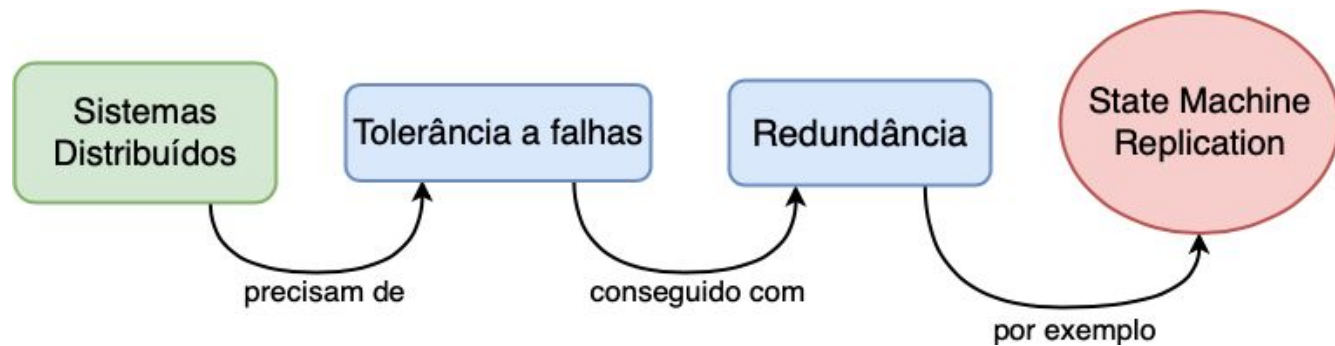
- Contexto
- Definição do problema
 - Objetivos
 - Trabalho Desenvolvido
 - Modelo e Pressupostos
- Abordagem
 - Arquitetura
 - Algoritmo 1 - Entrega Imediata
 - Algoritmo 2 - Entrega Diferida
- Resultados / Avaliação
 - Medições efetuadas

Contexto



*A distributed system is one in which the failure of
a computer you didn't even know existed can
render your own computer unusable.*

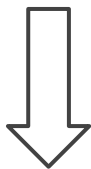
**Lamport, Maio de 1987
(num email)**





State Machine Replication

- abordagem genérica de desenvolvimento de sistemas tolerantes a falhas
- replicação da máquina de estados em vários nós
- execução da mesma sequência de operações nas réplicas



Como garantir esta
propriedade?

Multicast Fiável
Atômico

Definição do Problema



Objetivos

- Projetar...
- Implementar...
- Avaliar...

Protocolo de Multicast Fiável baseado em UDP



Trabalho Desenvolvido

- Design e implementação de um algoritmo com entrega imediata
- Design e implementação de um algoritmo com entrega diferida
- Escolha de métricas e implementação das mesmas
- Design e implementação das simulações
- Análise dos dados recolhidos



Conceitos essenciais

- **Vista:** conjunto de processos/nós
- **Group Membership Service:** fornece informação acerca de que processos pertencem a um grupo
- **Group Communication Service:** fornece serviços de mensagens em grupo
- **Receção de uma mensagem:** receção da mensagem na interface de rede
- **Entrega de uma mensagem:** entrega da mensagem à camada de aplicação
- **Mensagem estável:** uma mensagem é estável para um processo se este souber que todos os outros processos da vista a receberam



Modelo e Pressupostos

- **Grupo:**
 - dinâmico
 - fechado
- **Falhas:**
 - crash
 - não há recuperação
- **Canal de Comunicação:**
 - UDP Multicast

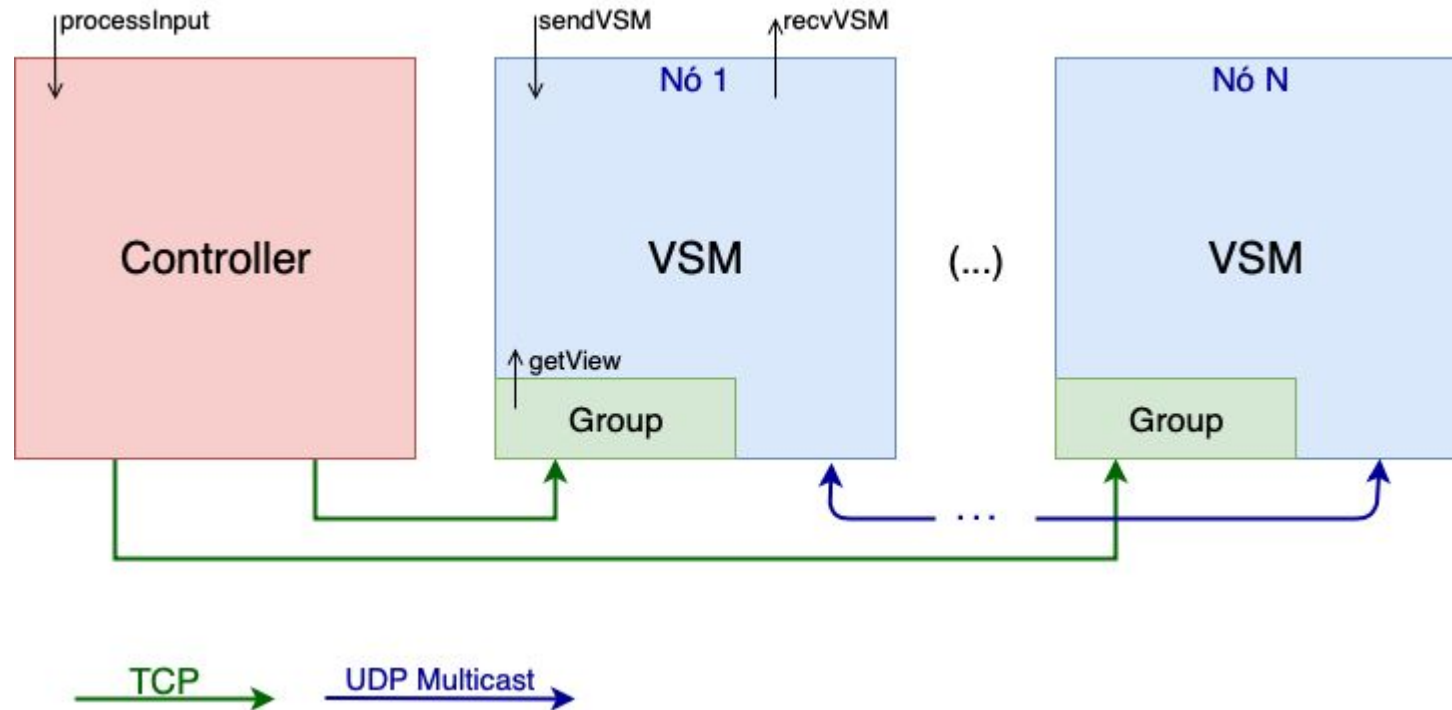


Propriedades

- **Virtual Synchrony:** se os processos p e q mudam da vista V para a V' , então entregam o mesmo conjunto de mensagens na vista V
- **Self Delivery:** se um processo correto p faz multicast de uma mensagem m , então entrega m
- **Ordem:** não ordenado

Abordagem

Arquitetura





Abordagem

Foram desenvolvidas duas alternativas:

1. Entrega imediata

Entrega mensagens sem as mesmas se encontrarem estáveis. Precisa de estabilizar as mesmas no processo de mudança de vista.

2. Entrega diferida

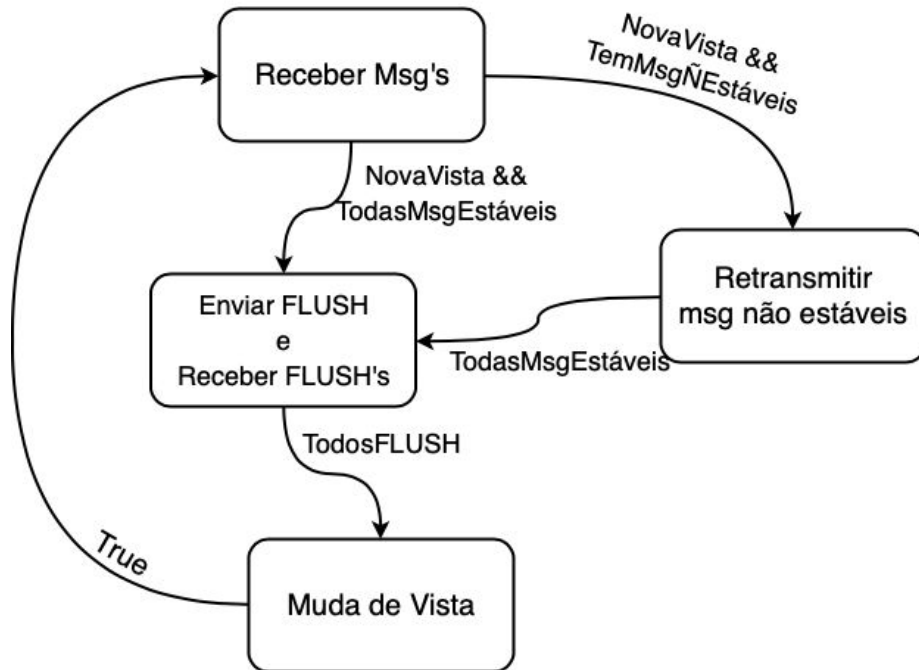
Entrega mensagens apenas após as mesmas estabilizarem, caso existam mensagens estáveis quando chega nova vista, então estas são descartadas.



Algoritmo 1 - Entrega Imediata

- Algoritmo baseado na implementação de Birman, mas sobre **UDP Multicast**
- Foram feitas as adaptações necessárias ao longo do desenvolvimento para garantir propriedade de **Virtual Synchrony**
- Para garantir a entrega de algumas mensagens, foram enviados Ack's, uma vez que o canal não é fiável

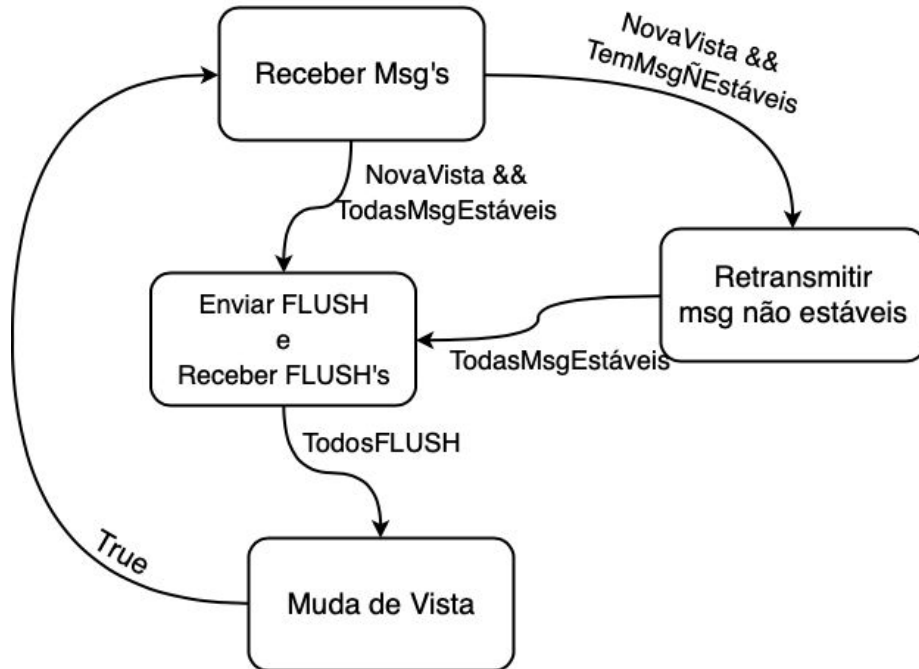
Algoritmo 1 - Entrega Imediata



Algoritmo executado por cada nó

1. Recebe uma nova vista para ser instalada
2. Caso tenha mensagens não estáveis
 - 2.1. Reenvia mensagens não estáveis com os respectivos Ack's (que já tinha recolhido no funcionamento normal)
 - 2.2. Espera respetivos Ack's (e verifica se entretanto a msg ficou estável)
3. Quando todas as mensagens estão estáveis
 - 3.1. Envia FLUSH
 - 3.2. Dá Ack aos FLUSH's que vai recebendo dos outros nós
 - 3.3. Espera que cheguem os FLUSH's de todos os processos e todos os ACK's ao seu FLUSH
4. Quando chegarem todos os FLUSH's e ACK's, instala a nova vista

Algoritmo 1 - Entrega Imediata



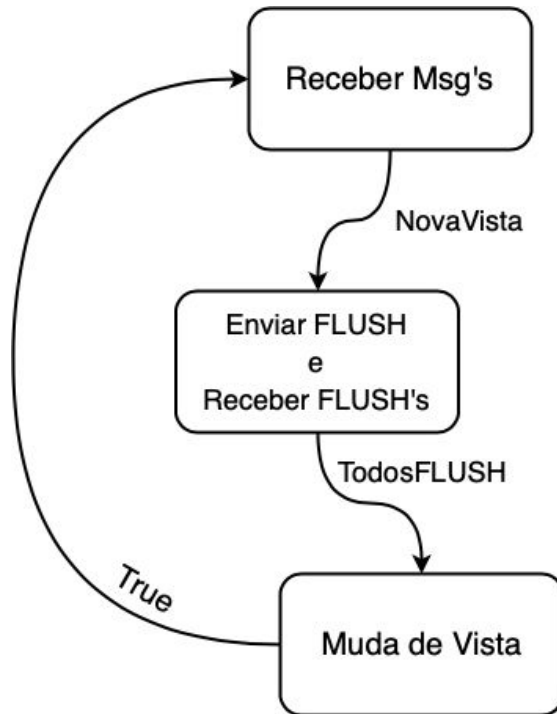
Mensagem de Flush contém as mensagens estáveis desse nó, para que os outros nós fiquem a conhecer o seu estado.



Algoritmo 2- Entrega Diferida

- Mensagens são entregues só depois de se tornarem estáveis
- Mensagens que se encontrem não estáveis durante a mudança de vista são descartadas
- Propriedade de self-delivery não garantida

Algoritmo 2 - Entrega Diferida



Flush com Mensagens entregues

Cada nó recebe flush de todos os outros e descobre que mensagens foram recebidas por todos. Depois, entrega as mensagens que se tornam estáveis e descarta as restantes.

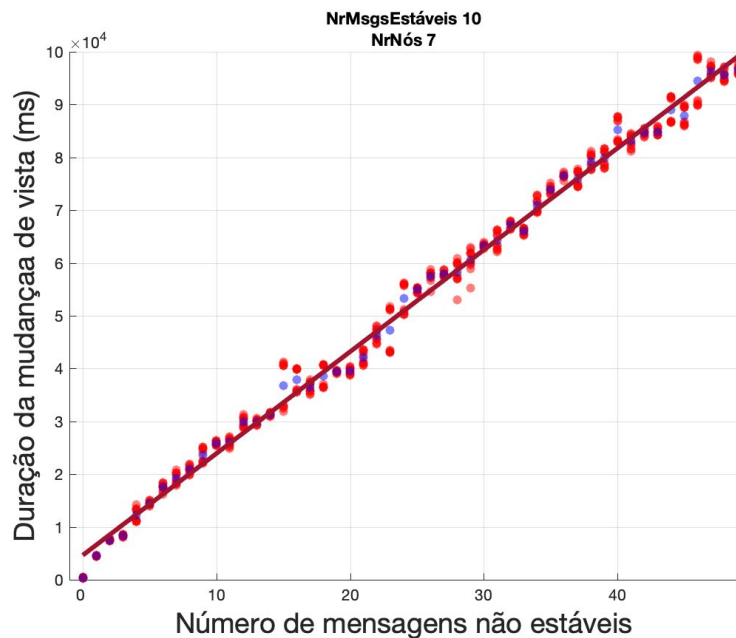
Resultados/Avaliação



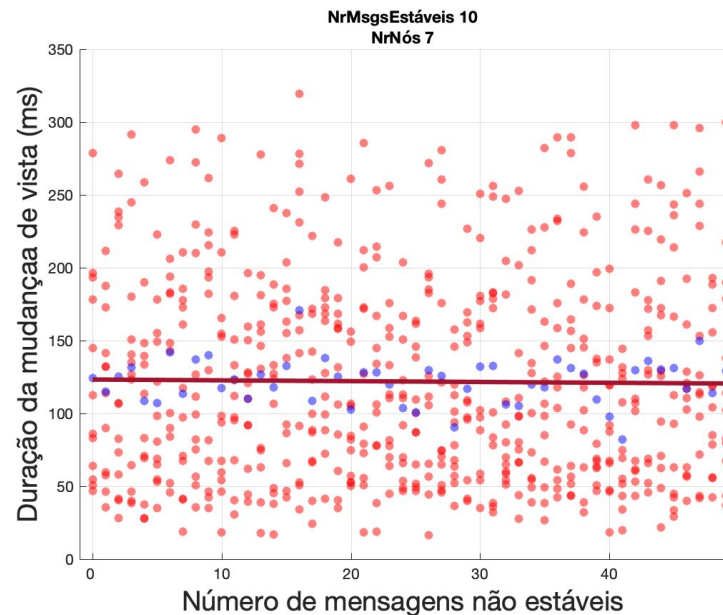
Medições efetuadas

- Métrica para comparação:
 - Tempo de mudança de vista
- Parâmetros:
 - N° de mensagens estáveis
 - N° de mensagens não estáveis
 - N° de nós

Medições efetuadas



Algoritmo 1

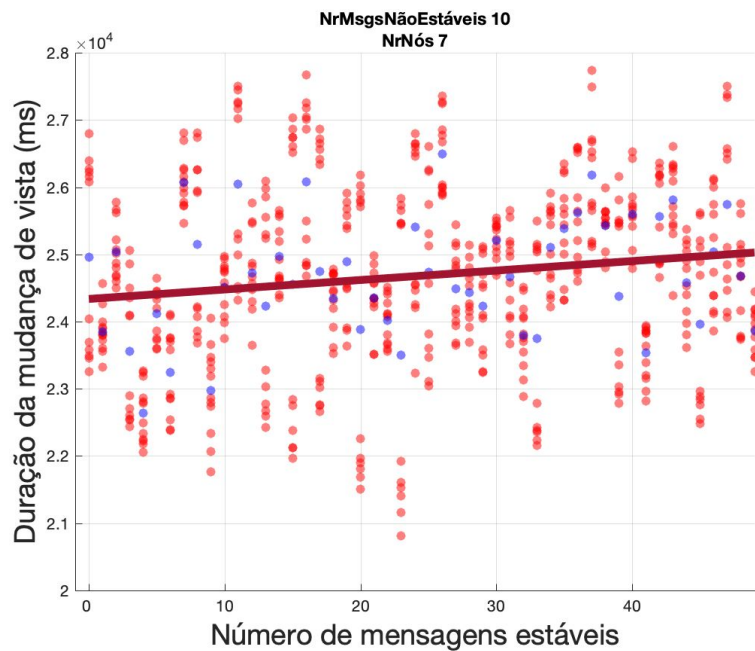


Algoritmo 2

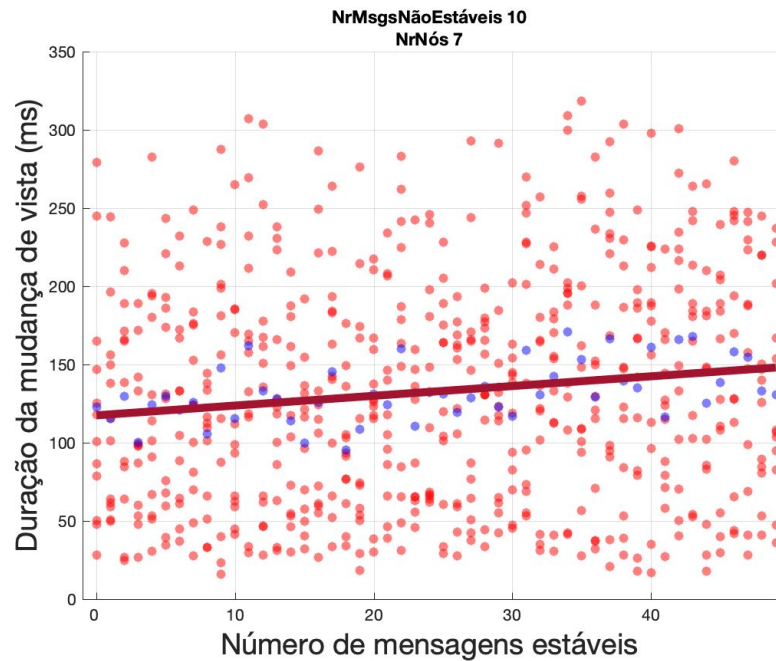


Algoritmo 2

Medições efetuadas



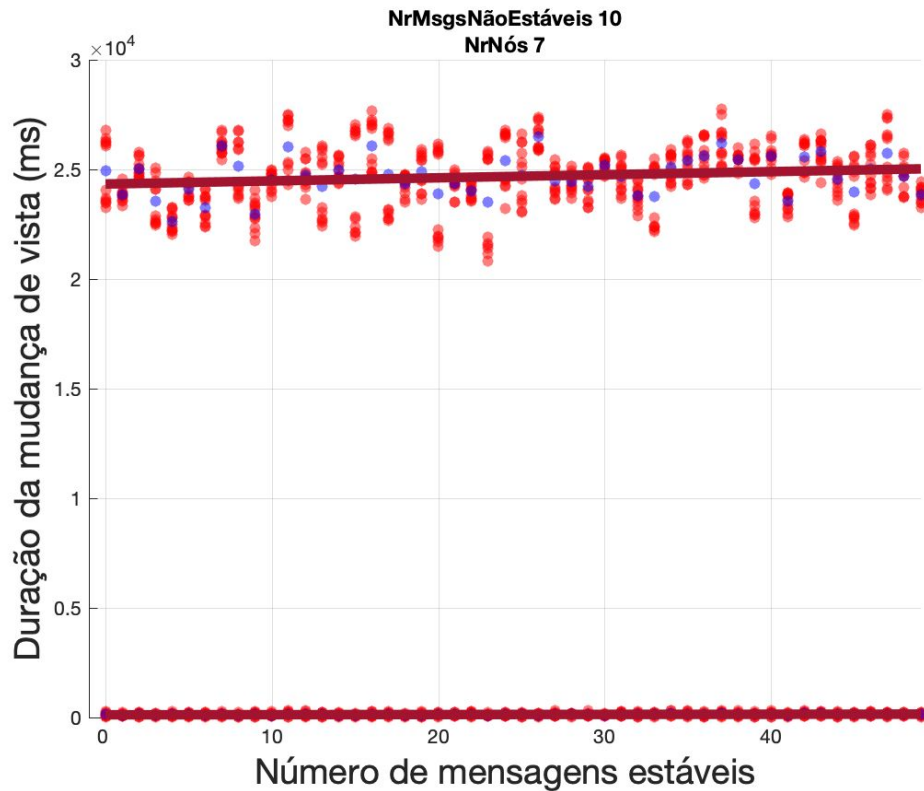
Algoritmo 1



Algoritmo 2



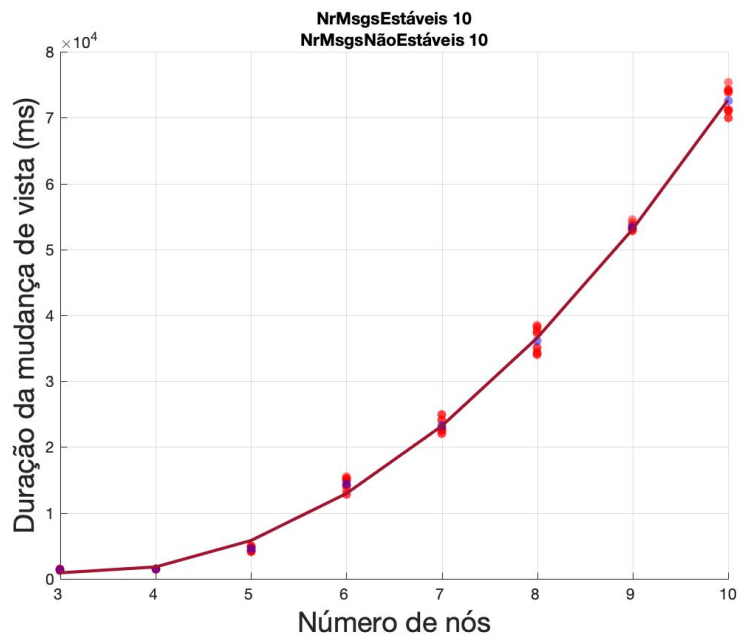
Medições efetuadas



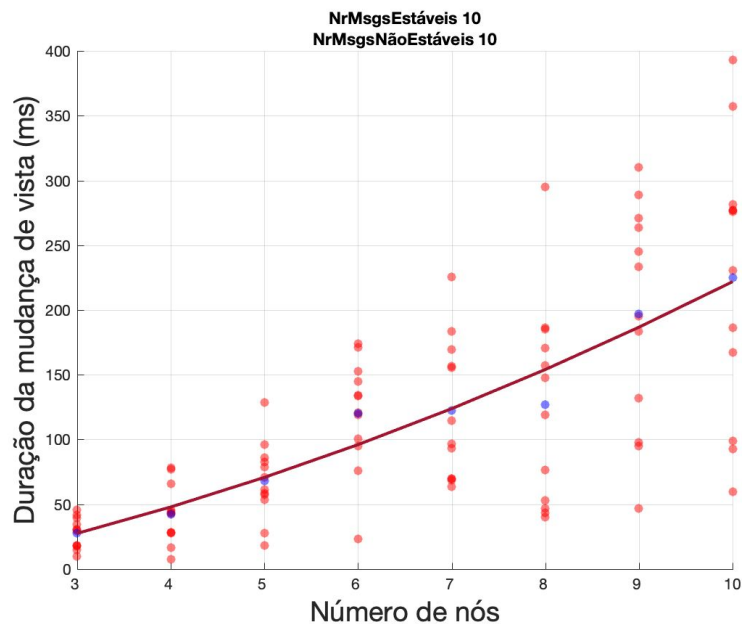
Algoritmo 1

Algoritmo 2

Medições efetuadas



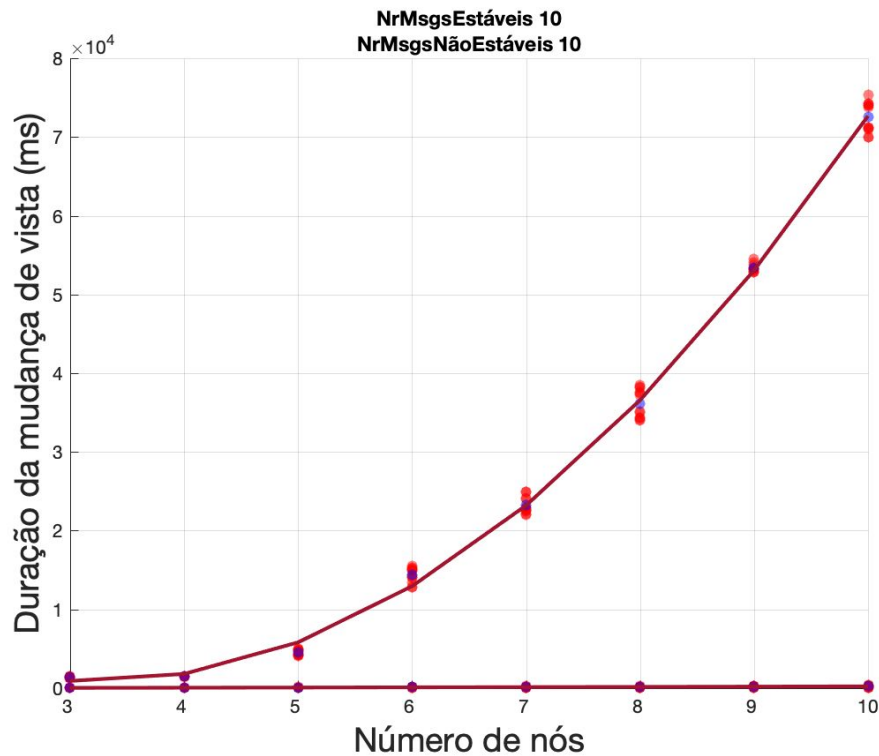
Algoritmo 1



Algoritmo 2



Medições efetuadas



Algoritmo 1

Algoritmo 2

Conclusão

Discussão & Questões





Referências

- Birman, K., Schiper, A., & Stephenson, P. (1991). Lightweight causal and atomic group multicast. *ACM Transactions on Computer Systems (TOCS)*, 9(3), 272-314.