Byzantine Consensus and Semantic Gossip: Scaling Decentralized Systems.

Ricardo Ferreira Guimarães - PPGCC
Pontifícia Universidade Católica do Rio Grande do Sul

Orientador: Prof.Dr. Fernando Luís Dotti

ricardo.guimarães@edu.pucrs.br

Apresentação Pessoal

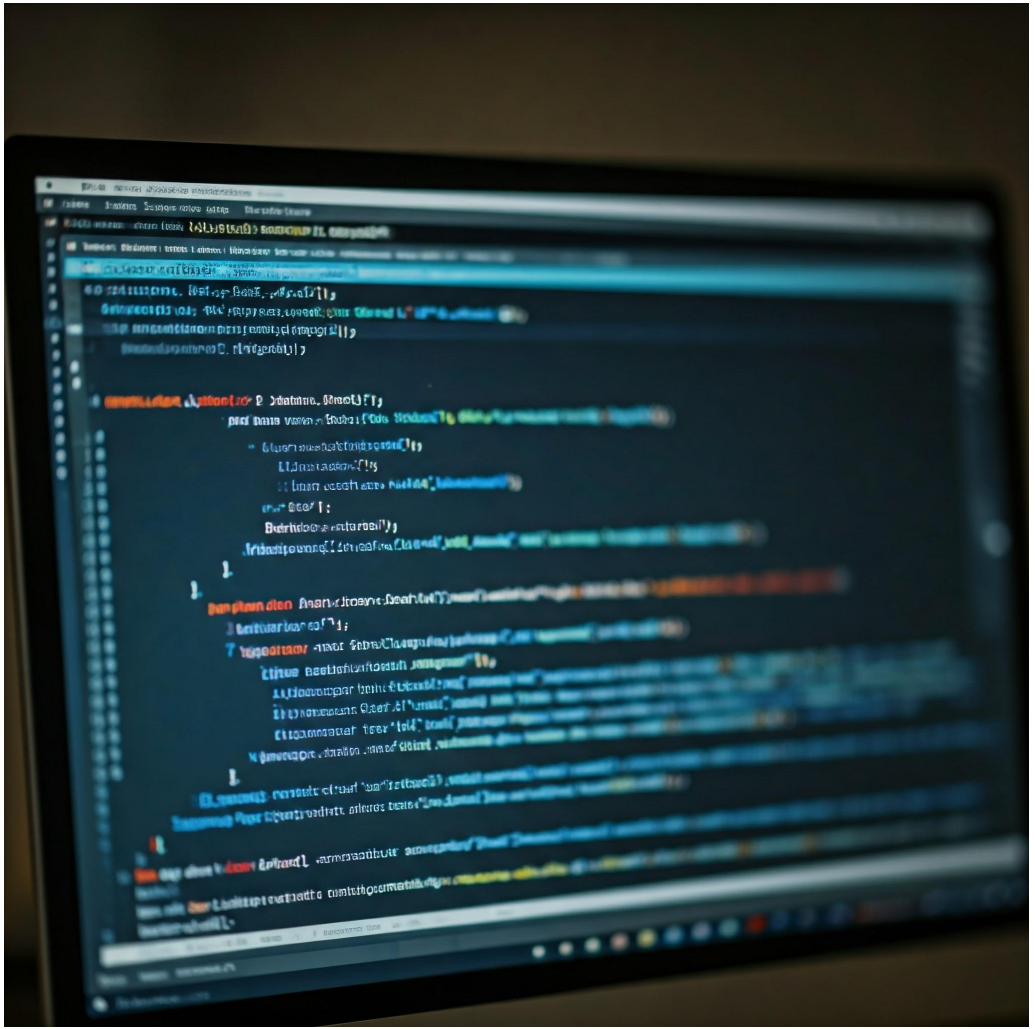
- Natural de Porto Alegre, Rio Grande do Sul, Brasil.
- 29 anos.
- Mestrando em Ciência da Computação na PUCRS.
- Orientador: Prof. Dr. Fernando Luís Dotti.

Minha Trajetória Acadêmica e Profissional

- Formado em Filosofia pela UFRGS
- TCC na área de Lógica Proposicional
- 4 anos de experiência como Desenvolvedor de Software
- Atuação no Ramo Bancário: Pix e Cartões

Meus Conhecimentos Técnicos

- Java
- Spring Boot
- SQL
- Microsserviços
- REST APIs
- CI/CD
- Kubernetes



Colaboração com a Informal Systems

- Cooperação entre o grupo de pesquisa do Prof. Dotti (PUCRS) e a Informal Systems.
- Informal Systems: ex-mantenedora do CometBFT (Tendermint em Go), desenvolvedora do Malachite (Tendermint em Rust).
- Objetivo: medir a performance do algoritmo.

Gossip Semântico em Tendermint

01 — Produção de artigo com Prof. Dotti e Prof. Pedone.

02 — Tema: Gossip Semântico na camada de comunicação do Tendermint.

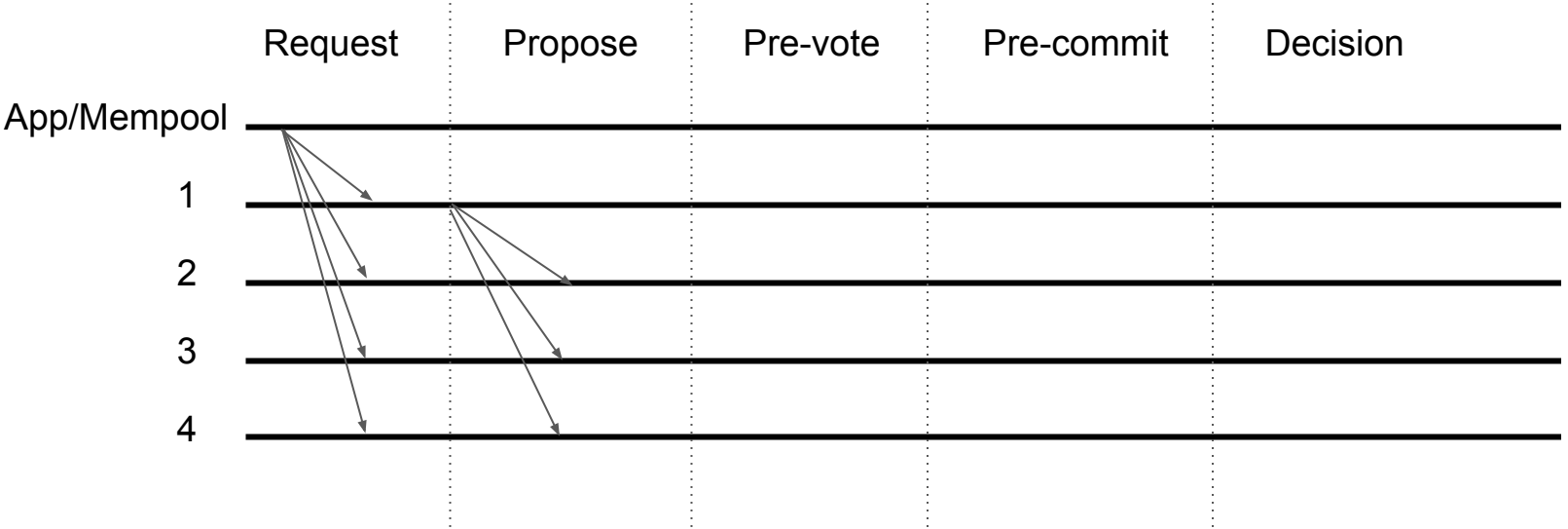
03 — Fruto de intercâmbio em Lugano, Suíça, com o grupo do Prof. Pedone.

O que é Tendermint?

- Algoritmo de consenso BFT (Byzantine Fault Tolerance).
- Garante replicação segura e consistente.
- Tolerante a falhas bizantinas: até ⅓ dos nós podem falhar ou agir maliciosamente.
- Utilizado em sistemas distribuídos.
- Ideal para blockchains e outras aplicações onde a confiança é essencial.

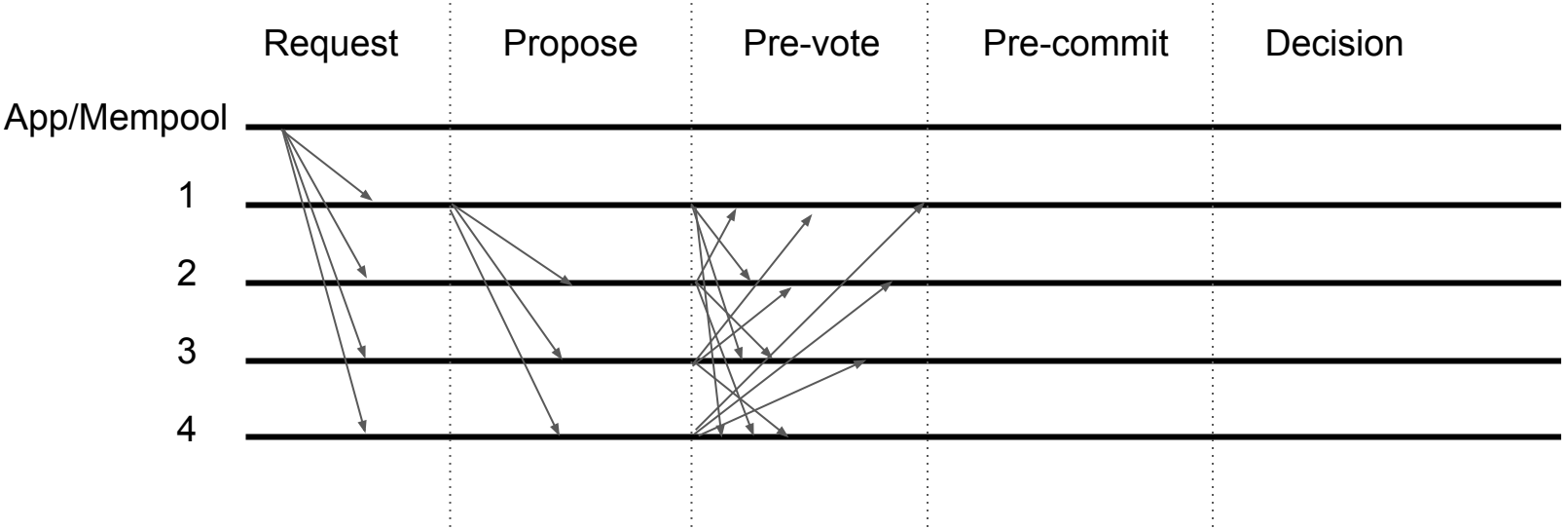
Propose

- Fase 1 (Proposta): Um validador é escolhido para propor um bloco em uma nova altura, enviando uma mensagem "Proposta" para todos os outros validadores.



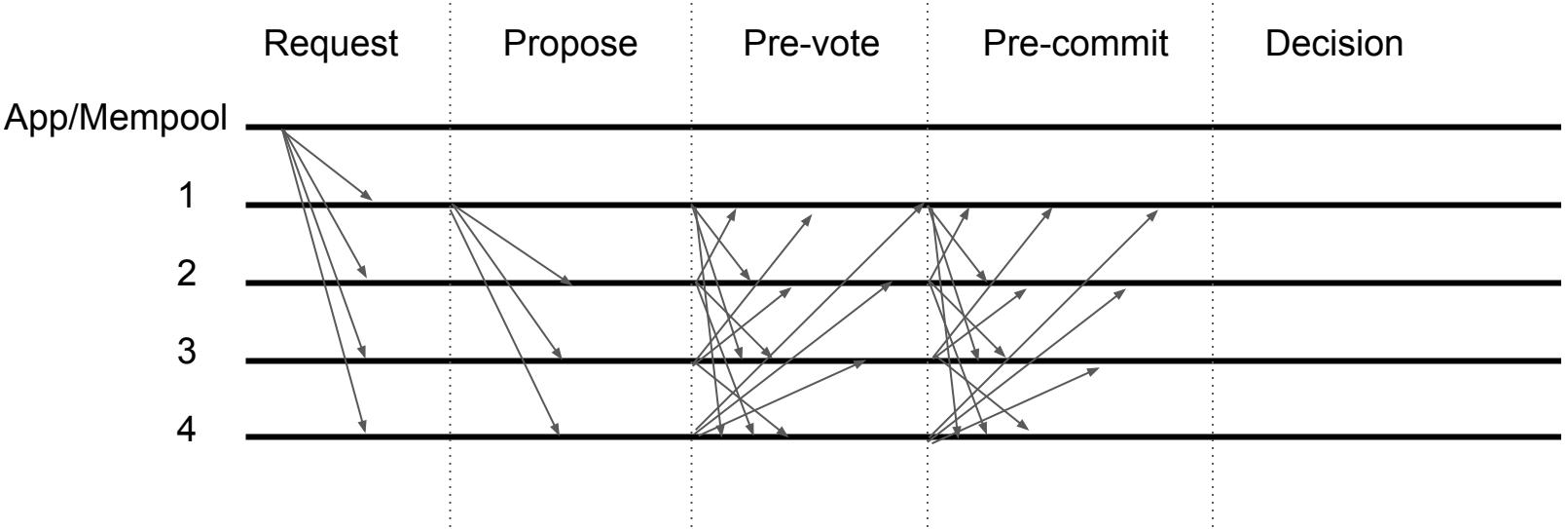
Pre-vote

- Fase 2 (Pre-voto): Cada validador verifica a proposta e envia uma mensagem "Pre-voto" indicando aprovação ou rejeição.



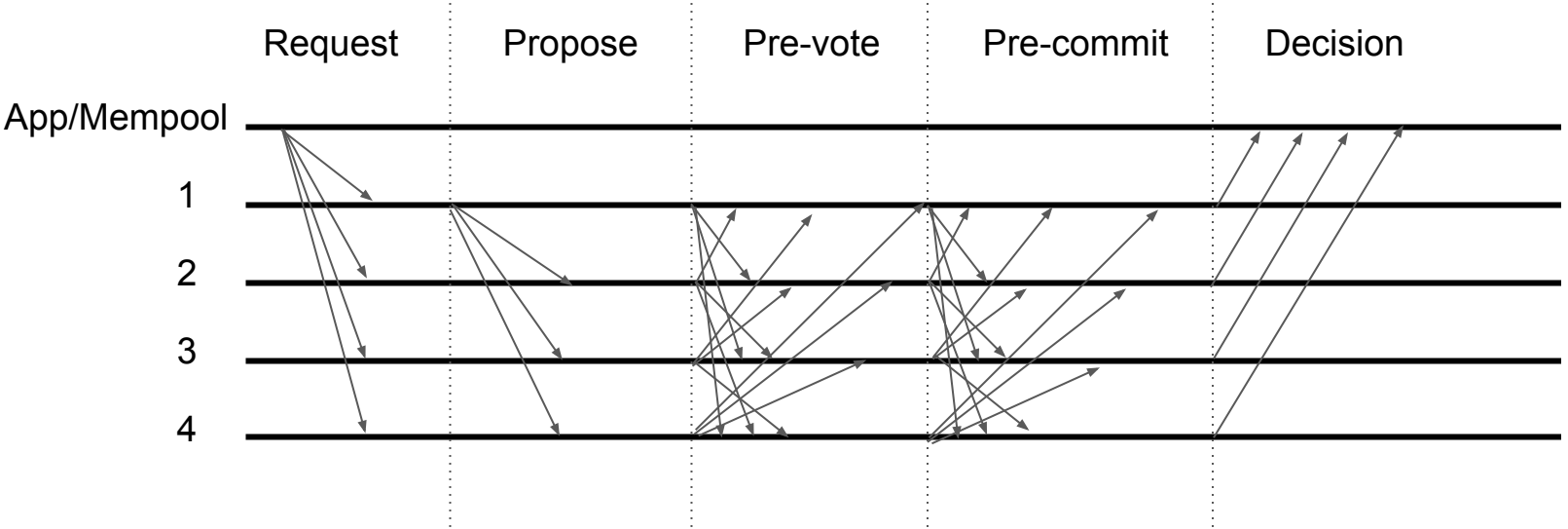
Pre-commit

- Fase 3 (Pre-commit): Os validadores trocam mensagens "Pre-commit" confirmando seus Pre-votos e verificando se há um quorum.



Decision

- Fase 4 (Decision): Se houver um quorum, os validadores enviam mensagens "Commit" confirmando o bloco e o adicionam ao blockchain.



Mensagens e Redundância no Gossip

- A comunicação via gossip ocorre por meio da troca de mensagens entre os nós.
- Cada nó envia mensagens periodicamente para um pequeno subconjunto de nós vizinhos.
- As mensagens recebidas são retransmitidas para outros vizinhos, propagando a informação pela rede.
- A redundância de mensagens é inerente ao protocolo gossip, garantindo a entrega da informação mesmo com perdas de mensagens ou falhas de nós.
- O número de mensagens trocadas em uma rede gossip pode ser elevado, especialmente em redes grandes ou com alta frequência de envio de mensagens.

Consenso + Gossip e redundância

Propostas:

Filtro:

se um nodo já repassou ⅔ N votos a vizinhos,
precisa repassar outros adicionais ?

Agregação:

concatenar todas mensagens a enviar a todos vizinhos em uma ?

overhead de processamento(validação de assinatura criptográfica)

Gossip Semântico

Introduz vulnerabilidades ao consenso ?

Filtro:

como mensagens são omitidas,

não prejudica safety

como mensagens omitidas são além das necessárias para consenso,

não prejudica liveness

Agregação:

não há modificação na informação trafegada,

apenas concatenação de mensagens

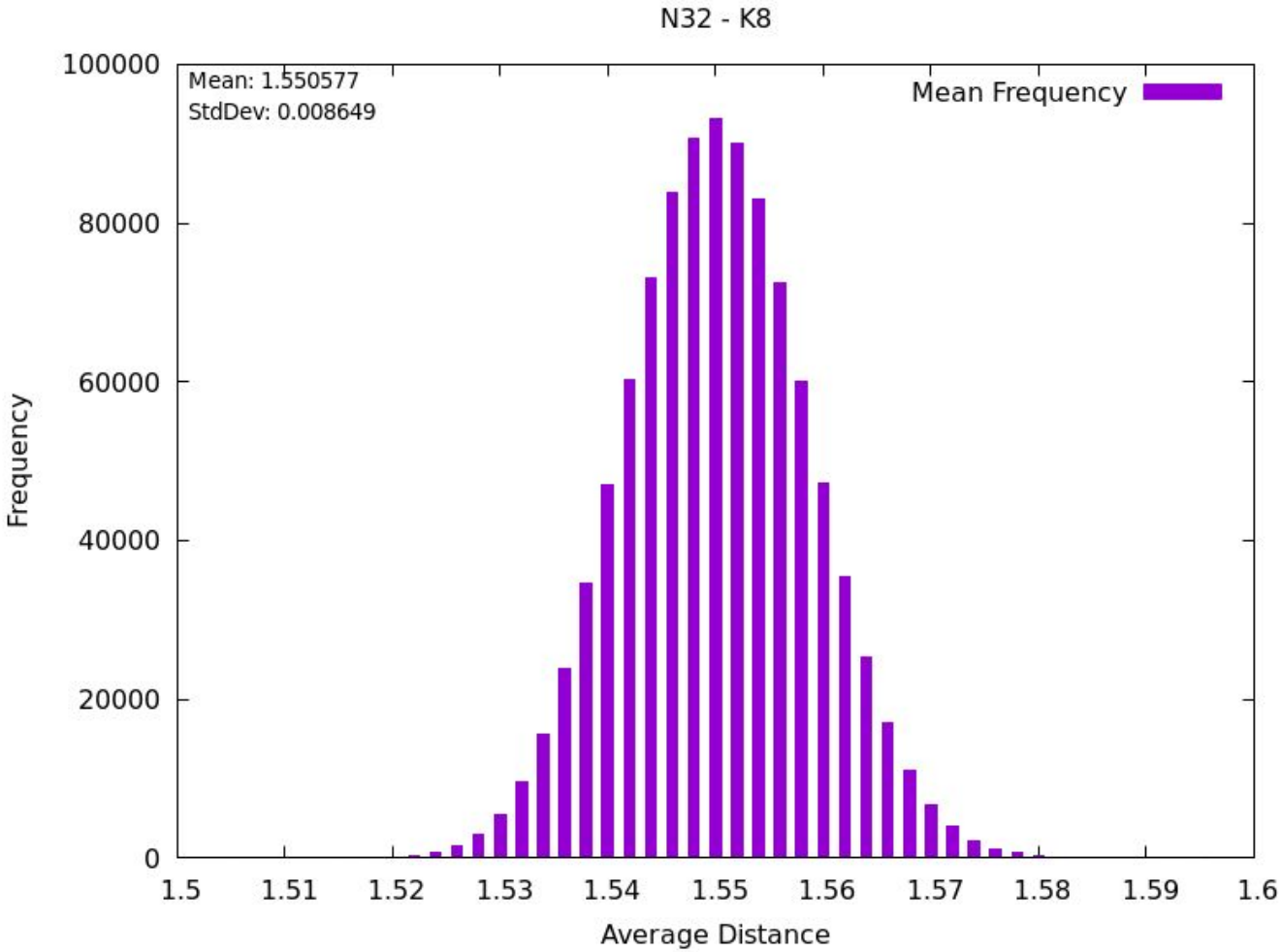
consenso processa mesmas mensagens

Avaliações

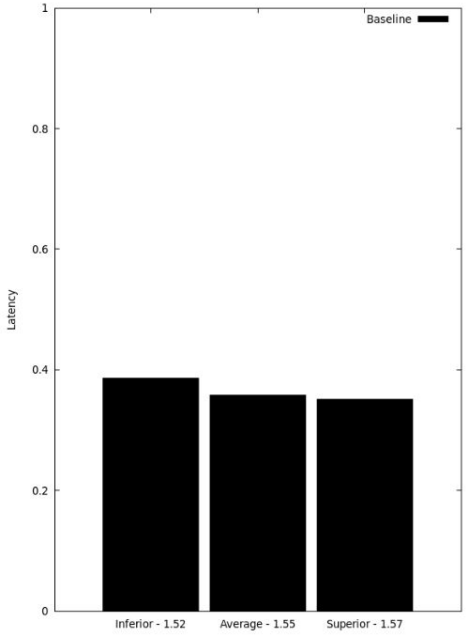
Experimentos:

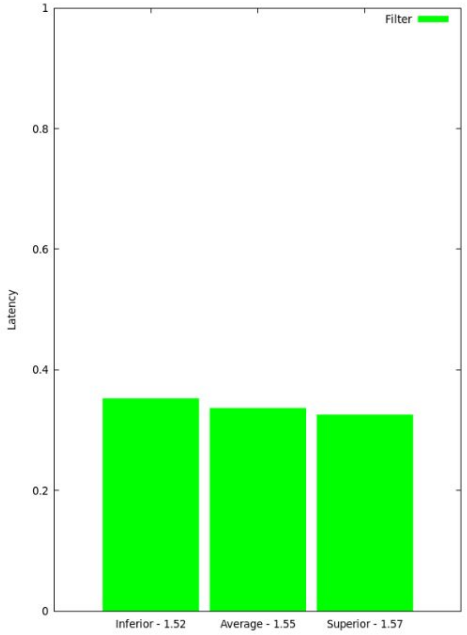
- topologias com 32 e 128 nodos
- geração de carga em cada nodo participante
- aumento da carga permitindo propostas concomitantes
  (mais mensagens trafegando simultaneamente)
- CloudLab
- Latências da AWS

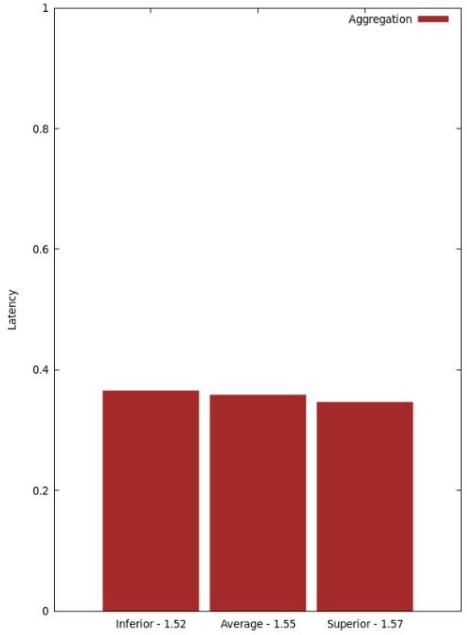
Resultados

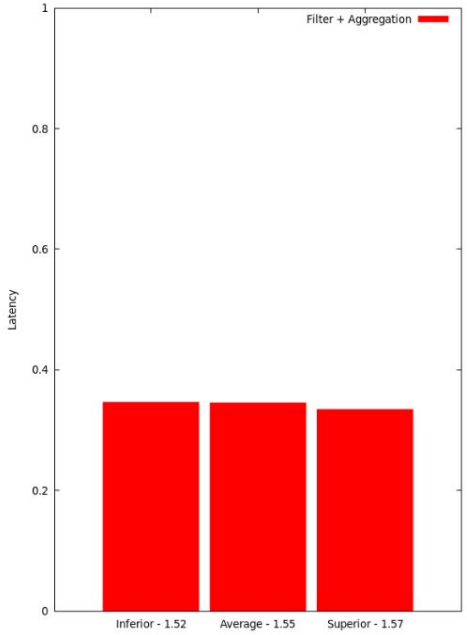

N32 - K8
Mean: 1.550577
StdDev: 0.008649
Mean Frequency
Frequency
100000
80000
60000
40000
20000
0
1.5
1.51
1.52
1.53
1.54
1.55
1.56
1.57
1.58
1.59
1.6
Average Distance

Latência entre topologias

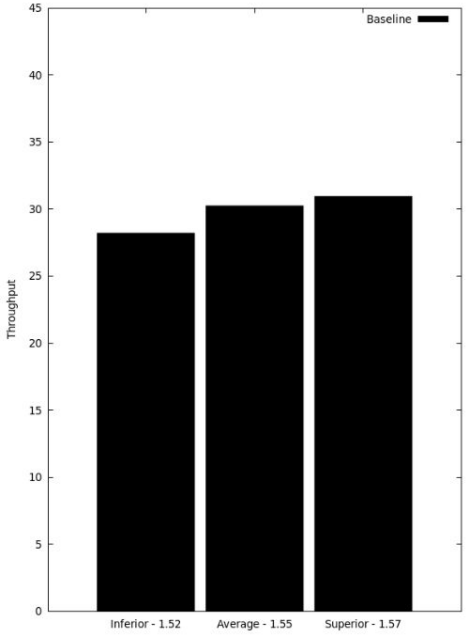


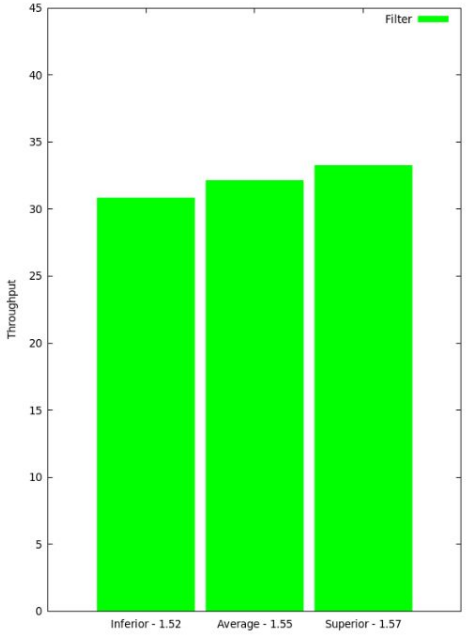


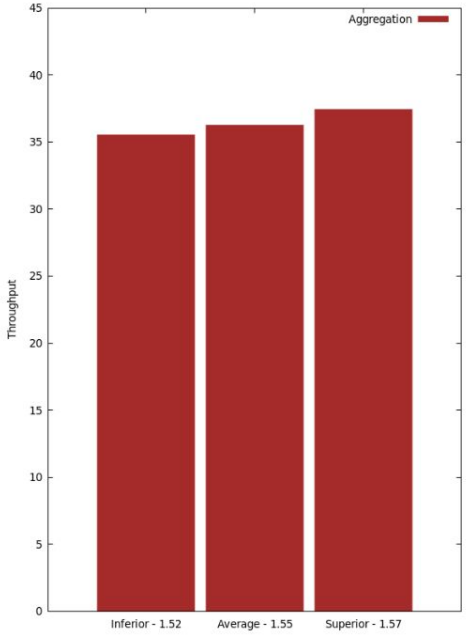


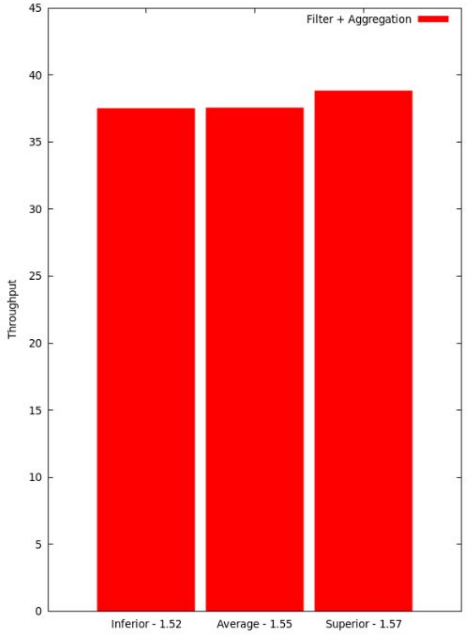


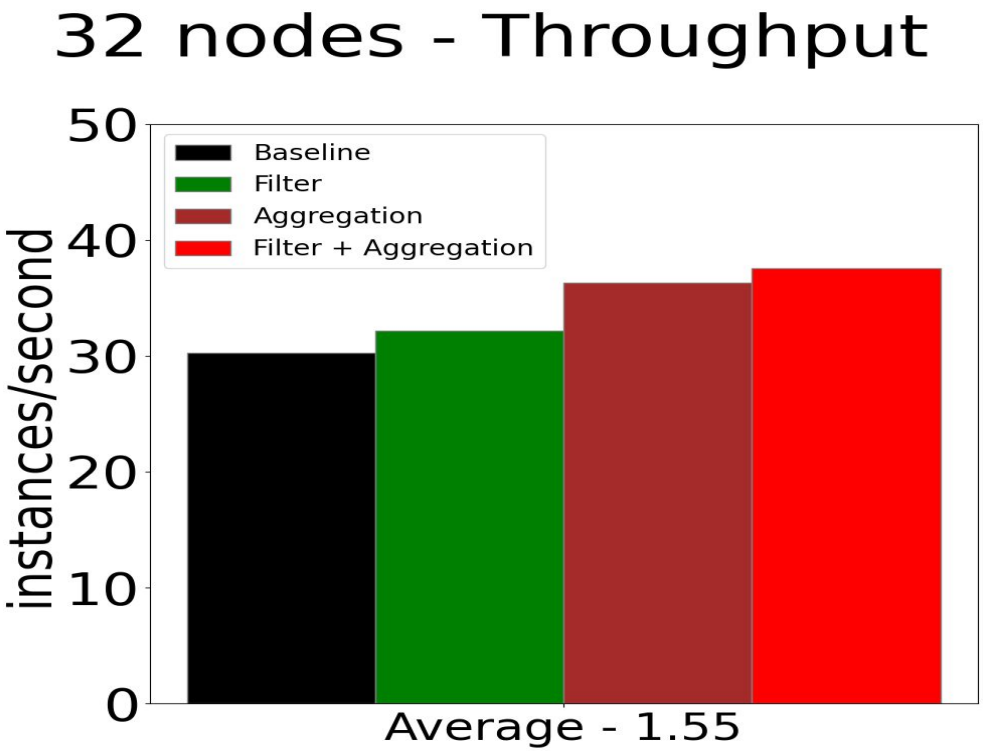
Vazão entre topologias

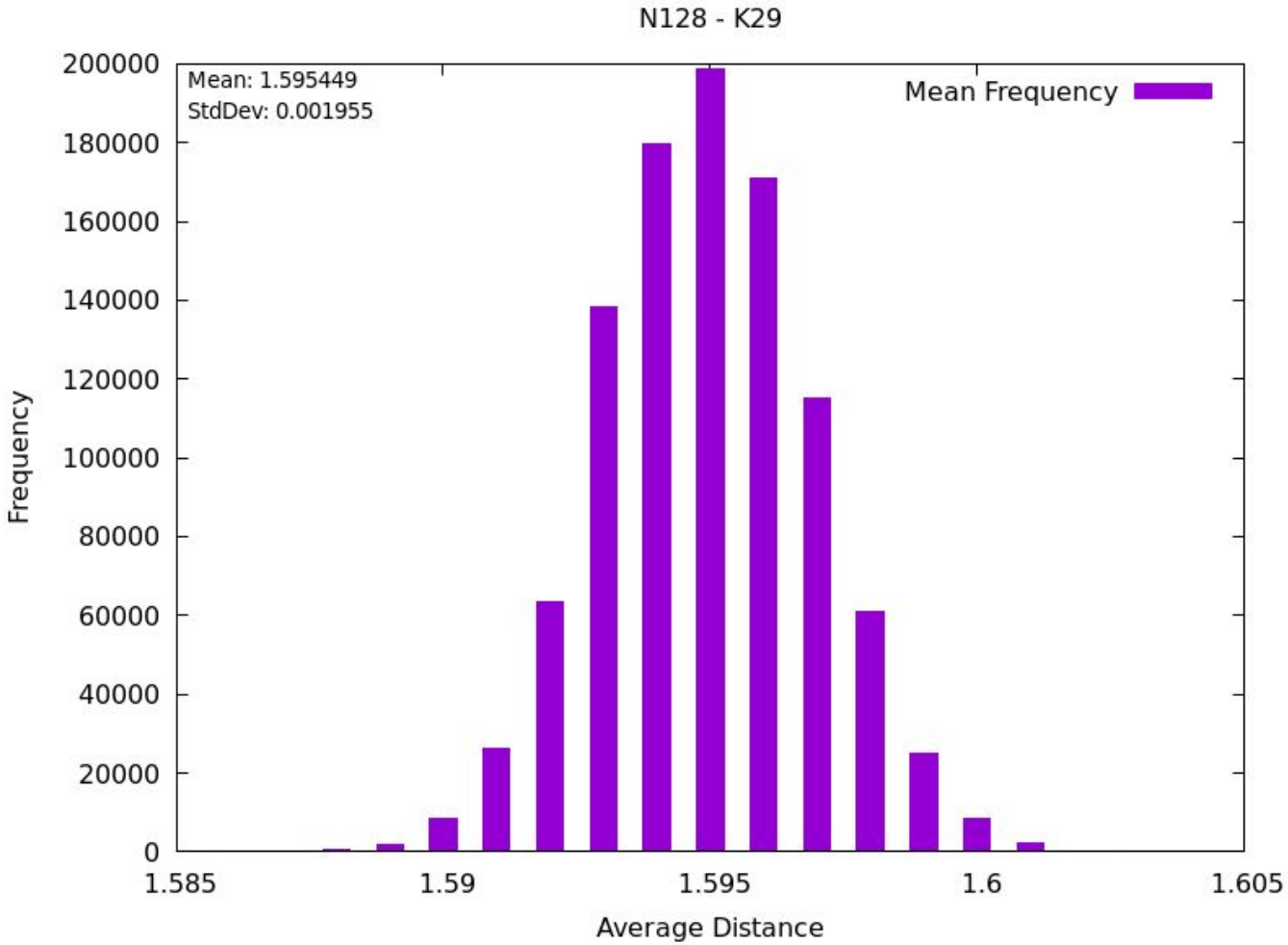




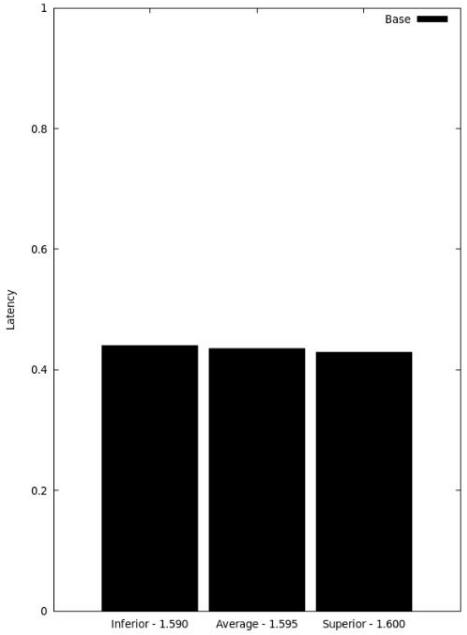


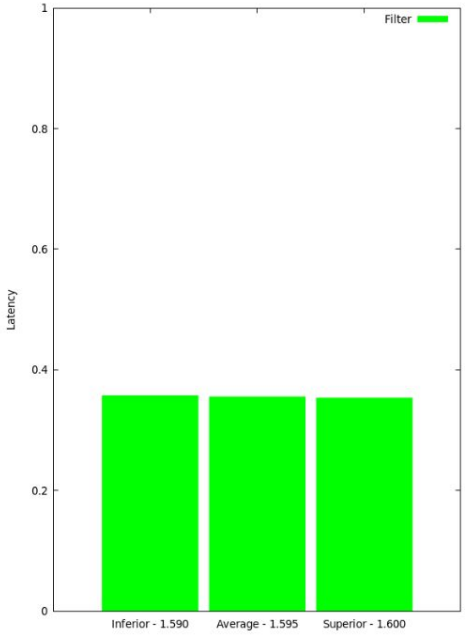


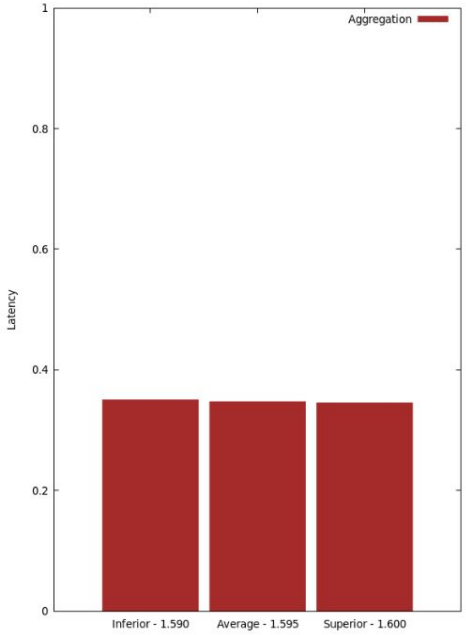

32 nodes - Throughput
instances/second
50
40
30
20
10
0
Baseline
Filter
Aggregation
Filter + Aggregation
Average - 1.55

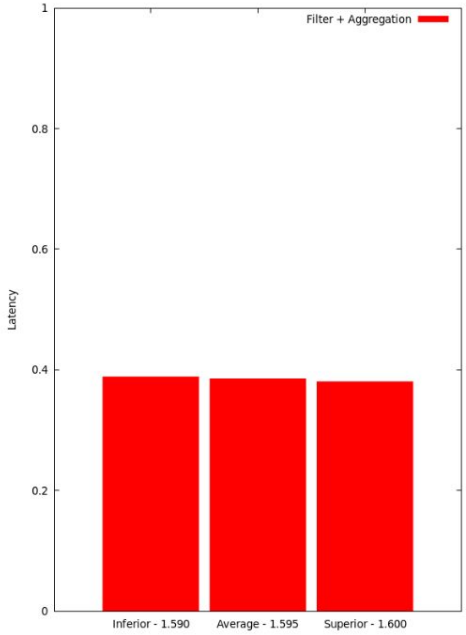

N128 - K29
Mean: 1.595449
StdDev: 0.001955
Mean Frequency
Frequency
200000
180000
160000
140000
120000
100000
80000
60000
40000
20000
0
1.585
1.59
1.595
1.6
1.605
Average Distance

Latência entre topologias

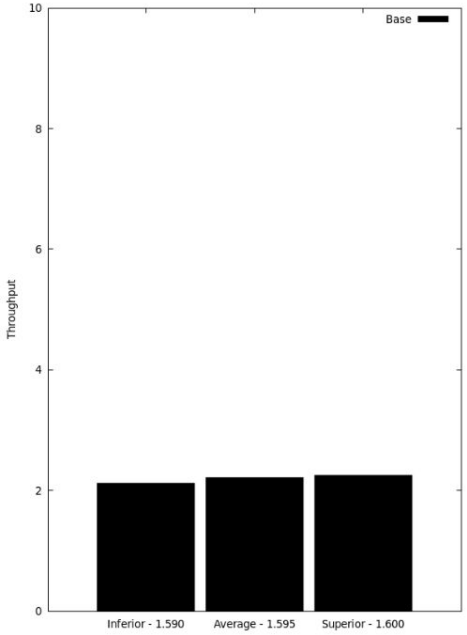


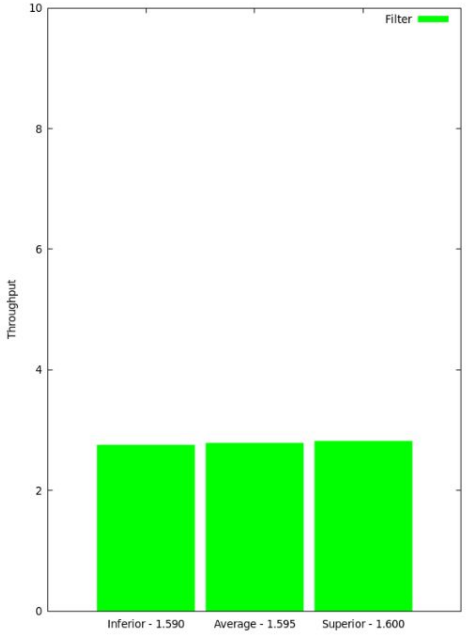


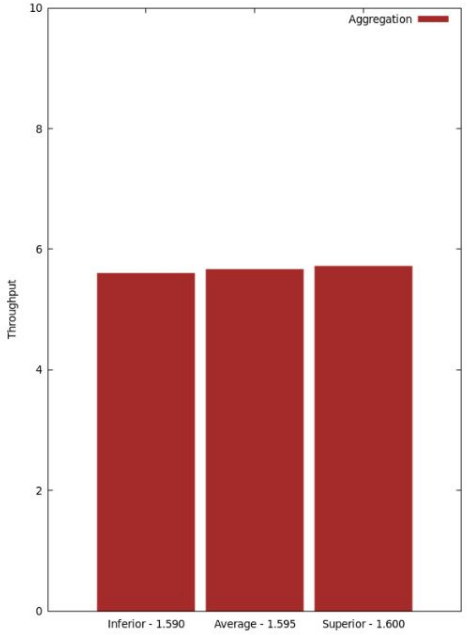


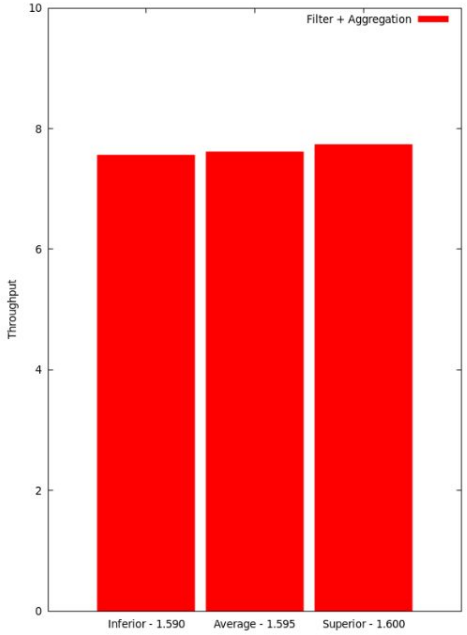


Vazão entre topologias









Percentual de mensagens filtradas.

Size in Nodes	Filtering	Filtering+ Aggregation
32	23.07%	23.17%
128	29.87%	28.81%

TABLE II

PERCENTAGE OF MESSAGES FILTERED OUT BY THE SEMANTIC FILTER MECHANISM AT THE BEST THROUGHPUT/LATENCY POINT.

Percentual de mensagens agregadas.

Size in Nodes	Aggregation	Filtering+ Aggregation
32	16.77%	13.01%
128	70.92%	71.76%

TABLE III
PERCENTAGE OF MESSAGES AGGREGATED AT THE BEST THROUGHPUT/LATENCY POINT.

Percentual de mensagens agregadas.

Size in Nodes	Estimated upper bound	Base-line	Filte-ring	Aggre-gation	Filter.+ Aggreg.
32	887	822,2	612,6	690,6	537,9
relative	1,079	1	0,74	0,83	0,65
128	13.166	12.805,4	8.876,3	3.613,1	2471
relative	1,028	1	0,69	0,25	0,19

TABLE IV

AVERAGE OF GOSSIP RECEIVED MESSAGES PER NODE, PER DECIDED TENDERMINT INSTANCE, AT THE BEST THROUGHPUT/LATENCY POINT.