



Teste de performance

Ismael Coral Hoepers Heinzelmann, Marcos Tomaszewski, Matheus Paulon Novais, Sérgio Bonini



Cenário de teste

- Tipo de broadcast → AB;
- Quantidade de nós → 4, 7 e 10;
- Perfis de falhas →
 - 0% de perda e 0% de corrupção;
 - 1% de perda e 0% de corrupção;
 - 1% de perda e 1% de corrupção;

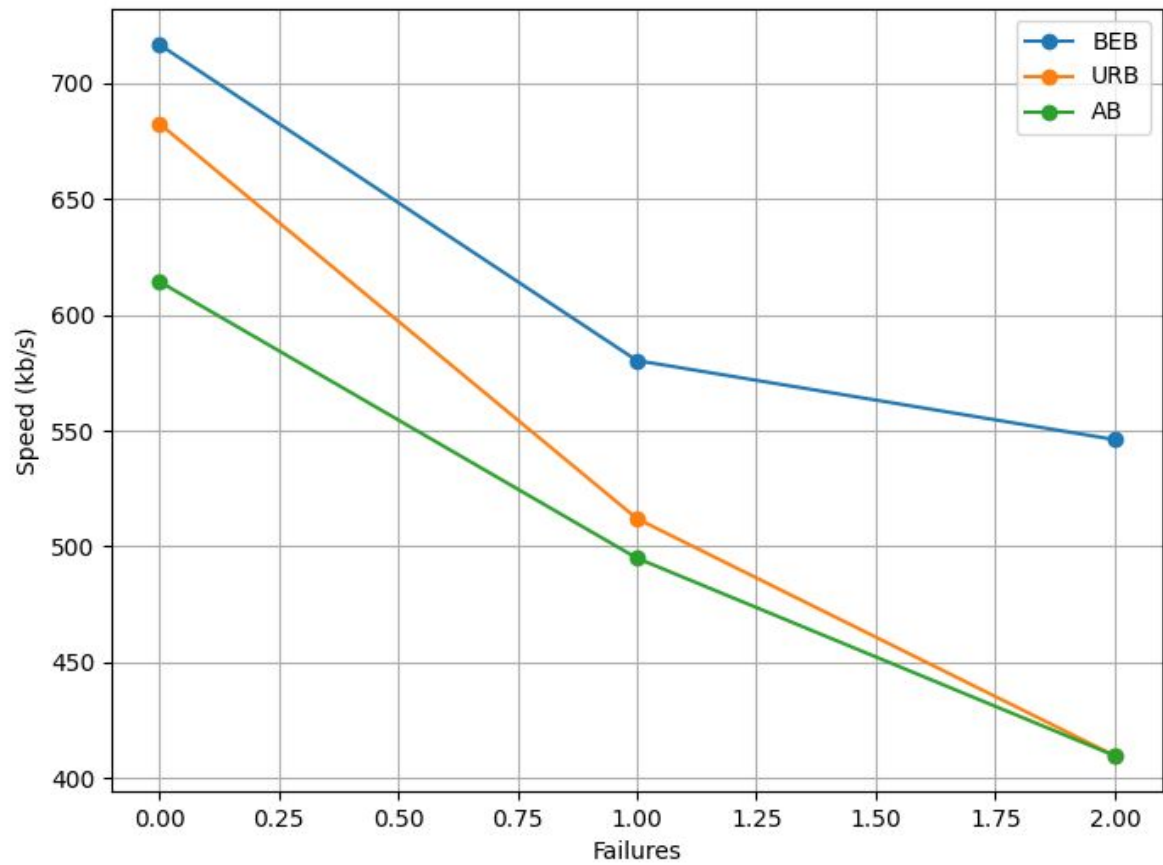


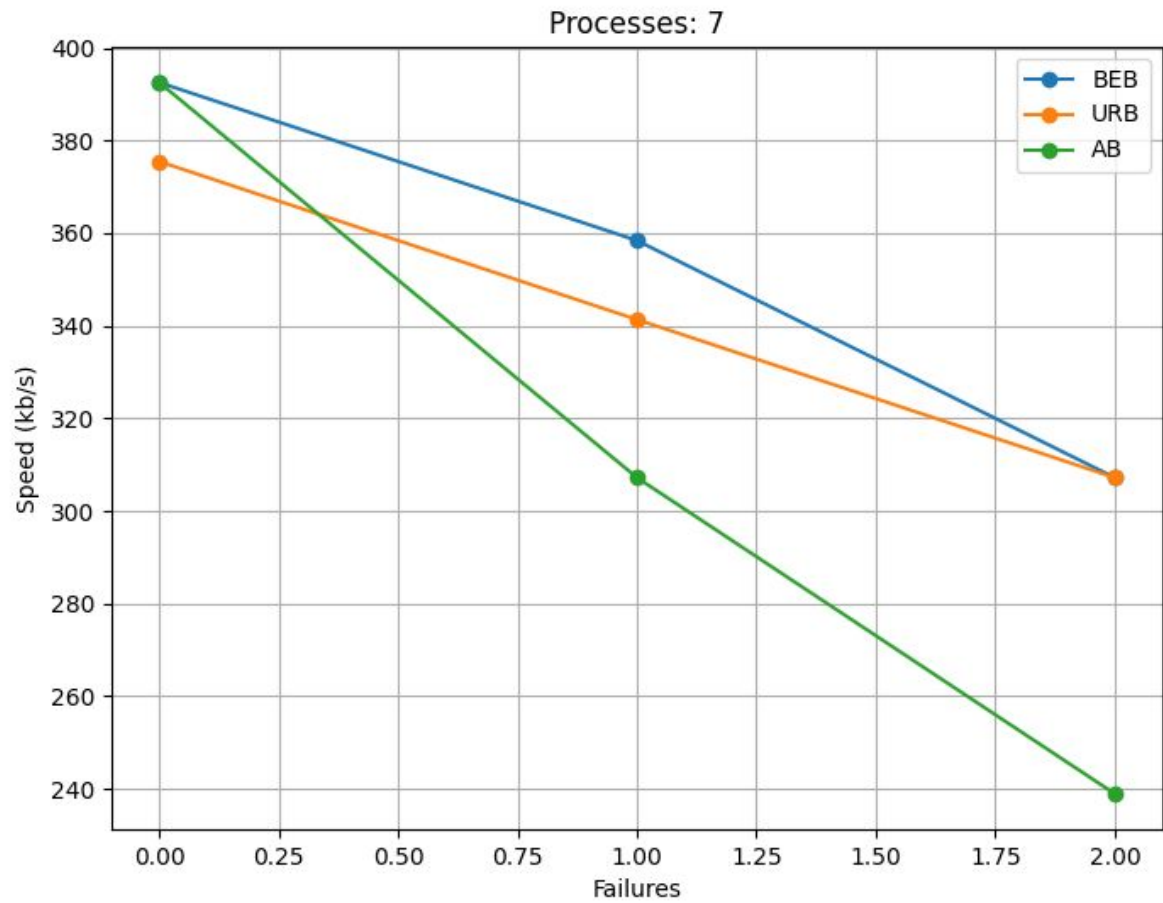
Cenário de teste

- Todos os nós enviam de dados aleatórios de teste em broadcast por 1 minuto;
- Os dados são pacotes de 50 KB;
- Todos os nós enviam estes dados e realizam medições de tempo;
- Para cada caso de teste, foram realizadas 5 execuções.

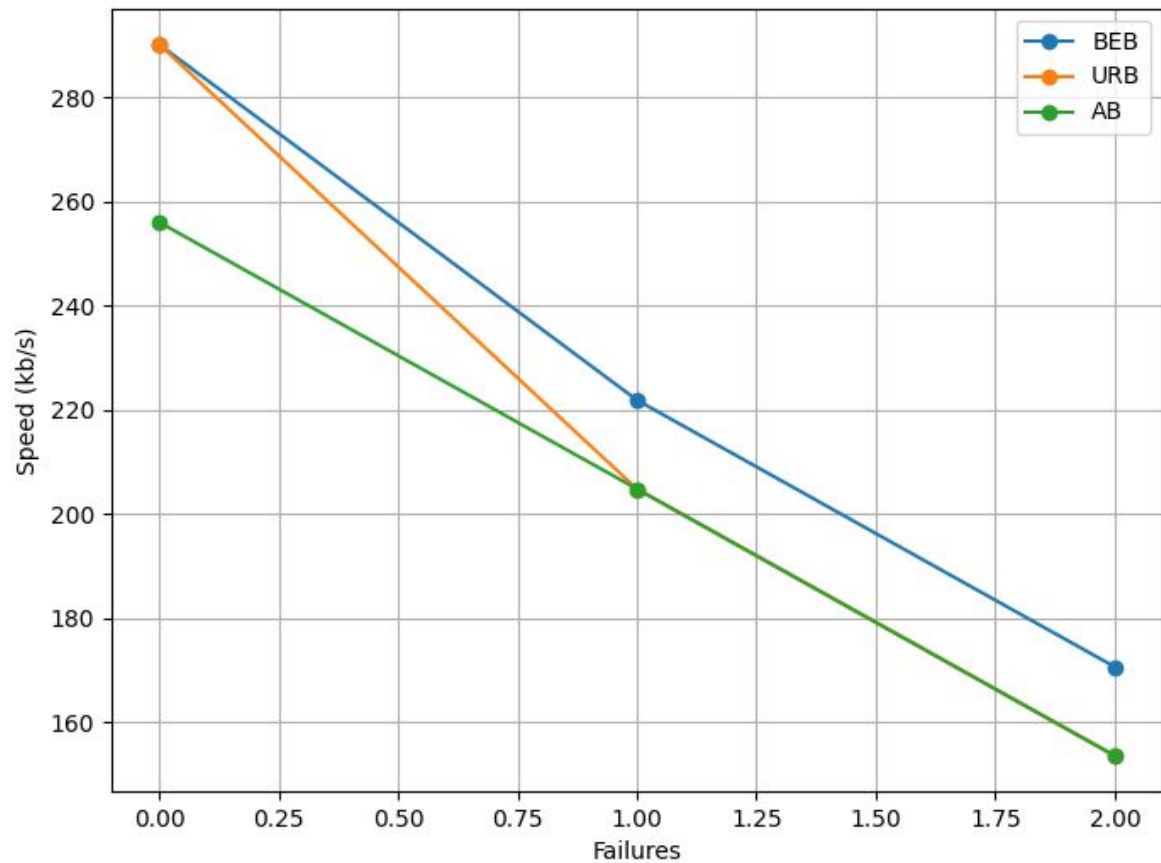
Resultados

Processes: 4

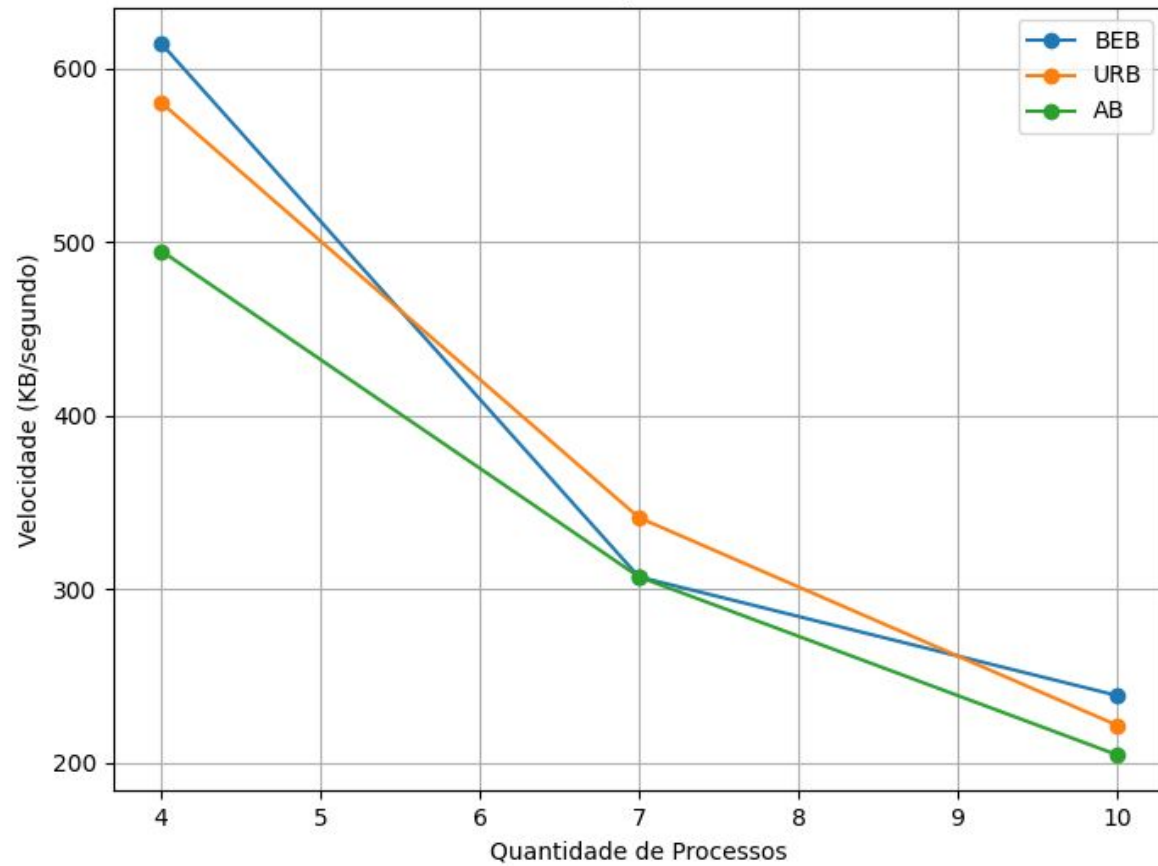




Processes: 10



Velocidade Média vs Quantidade de Processos





Conclusão

- Os dados experimentais mostram a relação entre a quantidade de processo e o *output*, sendo o fator mais significativo na redução do desempenho;
- Falhas também afetaram o desempenho do sistema, porém apenas de maneira muito mais limitada comparado ao fator número de processos;
- À medida que a quantidade de processos aumentava, a velocidade diminuía para todos os tipos de broadcast;
- Os *broadcasts* de tipo BEB e URB possuem desempenhos similares, já que sua única diferença é uma condição no momento da entrega;
- O broadcast do tipo AB teve velocidades inferiores graças a lógica de canal do grupo, limitando a velocidade de transmissão de múltiplos nós;



Conclusão

- Algumas melhorias podem ser implementadas aumentando drasticamente a velocidade do sistema:
 - Timeout dinâmico;
 - Utilização de datagramas de controle para *heartbeats*, reduzindo saturação da rede (como feito pelos colegas);
 - Utilização de lógica adicional para datagramas de controle, ofertando informações adicionais, reduzindo a necessidade de retransmissão de diversos datagramas de controle, reduzindo saturação da rede.
 - Utilização de estruturas de dados melhores, evitando iterações excessivas;
 - Aumentar paralelismo do sistema, onde a recepção pode ser feita por uma *pool* de threads.