



UNIVERSIDADE FEDERAL DO AMAZONAS
FACULDADE DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

WILLIAM CASTRO DA ROSA

**QoN++: ROTEAMENTO POR QUALIDADE DOS NÓS COM REDUÇÃO DE
SOBRECARGA EM DTN**

Orientador: Prof. Dr. Celso Barbosa Carvalho
Coorientador: Prof. Dr. José Ferreira de Rezende

MANAUS - AMAZONAS
OUTUBRO - 2020

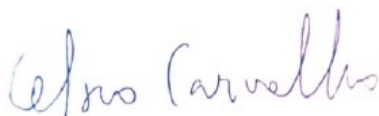
**QoN++: ROTEAMENTO POR QUALIDADE DOS NÓS COM REDUÇÃO DE
SOBRECARGA EM DTN**

WILLIAM CASTRO DA ROSA

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas, como requisito parcial para obtenção do título de Mestre em Engenharia Elétrica na área de concentração Controle e Automação de Sistemas.

Aprovada em 29 de outubro de 2020.

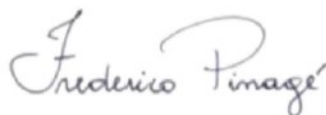
BANCA EXAMINADORA



Prof. Dr. Celso Barbosa Carvalho, Presidente
Universidade Federal do Amazonas



Prof. Dr. Edjair de Souza Mota, Membro
Universidade Federal do Amazonas



Prof. Dr. Frederico da Silva Pinagé, Membro
Instituto Centro de Tecnologia de Software

MANAUS - AMAZONAS

OUTUBRO – 2020

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

D111q Rosa, William Castro da
QoN++: Roteamento por qualidade dos nós com redução de sobrecarga em DTN / William Castro da Rosa .
2020 65 f.: il. color; 31 cm.

Orientador: Celso Barbosa Carvalho
Coorientador: José Ferreira de Rezende
Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal do Amazonas.

1. Redes Tolerantes a Atrasos e Desconexões. 2. Roteamento. 3. Qualidade dos Nós. 4. Sobrecarga. I. Carvalho, Celso Barbosa. II. Universidade Federal do Amazonas III. Título

AGRADECIMENTOS

A Deus, por nos permitir admirar e estudar a perfeição de Suas obras.

À família – César, Estrela e Allan – por ser o cerne da minha educação e a fonte dos melhores valores morais.

Aos amigos e professores do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Amazonas pelo incentivo em momentos de dificuldades.

Ao meu orientador, Celso Carvalho, pelas contribuições intelectuais na elaboração deste trabalho.

Ao meu coorientador, José Ferreira de Rezende, pelas reflexões e análises necessárias ao aprimoramento da pesquisa.

Aos membros da banca, os professores Edjair de Souza Mota e Frederico da Silva Pinagé, pelas sugestões aplicadas na versão final deste documento.

Esta pesquisa, conforme previsto no Art. 48 do decreto nº 6.008/2006, teve financiamento da Samsung Eletrônica da Amazônia Ltda, nos termos da Lei Federal nº 8.387/1991, através de convênio nº 004, firmado com o Centro de P&D em Eletrônica e Tecnologia da Informação da Universidade Federal do Amazonas - CETELI / UFAM; e foi apoiada por: Coordenação de Aperfeiçoamento Pessoal de Nível Superior - CAPES, Fundação de Pesquisa do Estado do Amazonas - FAPEAM no programa FAPEAM/PPP (EDITAL 004/2017), Fundo Setorial de Infraestrutura (CT-INFRA), MCT / CNPQ, SECTI/AM e Governo do Estado do Amazonas.

Resumo da Dissertação apresentada à UFAM como parte dos requisitos para a obtenção de grau de Mestre em Engenharia Elétrica.

QoN++: ROTEAMENTO POR QUALIDADE DOS NÓS COM REDUÇÃO DE SOBRECARGA EM DTN

William Castro da Rosa

Outubro / 2020

Orientador: Celso Barbosa Carvalho

Programa: Pós-Graduação em Engenharia Elétrica

A presente dissertação tem como objetivo reduzir a sobrecarga em redes tolerantes a atrasos e desconexões. Para esta finalidade, propõe-se o algoritmo QoN++, uma extensão do algoritmo QoN-ASW que utiliza o conceito de Qualidades dos Nós para ajustar a estratégia de encaminhamento de mensagens em uma rede e que considera as entregas de mensagens como eventos a serem notificados aos demais nós da rede. O simulador ONE é utilizado em conjunto com o arquivo de trace do evento Infocom06 para simular o algoritmo QoN++ e compará-lo com o desempenho de QoN-ASW. Para o projeto de experimentos, emprega-se o modelo Fatorial 2^k no estudo do efeito das interações de múltiplos fatores na sobrecarga na rede. Na análise de desempenho, variamos o tamanho de buffer dos nós de uma rede, o número de cópias de mensagens e seus intervalos de geração. Como contribuição deste trabalho, destaca-se o eficiente gerenciamento de recursos de buffer proporcionado pelo QoN++, oferecendo redução de sobrecarga em redes e aumento nas taxas de entrega de mensagens, ao custo de um pequeno acréscimo nos atrasos médios.

Palavras-chave: Redes tolerantes a atrasos e desconexões, roteamento, qualidade dos nós, sobrecarga.

Abstract of Dissertation presented to UFAM as part of the requirements for the
Master Degree in Electrical Engineering.

QoN++: QUALITY OF NODES-BASED ROUTING WITH REDUCED OVERHEAD IN
DTN

William Castro da Rosa

October / 2020

Advisor: Celso Barbosa Carvalho

Program: Post-Graduate in Electrical Engineering

This dissertation aims to reduce the overhead in Delay-Disruption Tolerant Networks. For this purpose, the QoN ++ algorithm is proposed - an extension of the QoN-ASW algorithm that uses the concept of Quality of Nodes to adjust the message forwarding strategy in a network, and considers message deliveries as events to be notified to other nodes in the network. The ONE simulator is used in conjunction with the Infocom06 trace file to simulate the QoN++ algorithm and compare its performance with QoN-ASW. For the experiments design, the 2^k Factorial model is used to study the effect of interactions of multiple factors on network overhead. In the performance analysis, we varied the buffer size of the nodes in a network, the number of message copies and their generation intervals. As a contribution of this work, the efficient buffer management provided by QoN++ stands out, offering a reduction in network overhead and an increase in message delivery rates, at the cost of a small increase in average delays.

Keywords: Delay and disruption tolerant networks, routing, quality of nodes, overhead.

SUMÁRIO

1. INTRODUÇÃO.....	13
1.1 Descrição do Problema.....	15
1.2 Motivação.....	15
1.3 Objetivos	16
1.3.1 Geral.....	16
1.3.2 Específicos	16
2. TRABALHOS RELACIONADOS.....	17
2.1 Roteamento Epidêmico.....	17
2.2 P _{Ro} PHET.....	18
2.2.1 Probabilidade de contato direto	18
2.2.2 Probabilidade transitiva	18
2.2.3 Fator de envelhecimento	19
2.3 Spray and Wait.....	20
2.3.1 Fase <i>Spray</i>	20
2.3.2 Fase <i>Wait</i>	22
2.4 QoN-ASW: Adaptive Spray and Wait Based on Quality of Nodes in DTN	22
2.4.1 Capacidade de manipulação de mensagens.....	22
2.4.2 Probabilidade de Entrega	24
2.4.3 Qualidade dos Nós	25
2.4.4 O Algoritmo do QoN-ASW	25
2.5 Outros Trabalhos	27
3. ALGORITMO QoN++.....	29
4. PROJETO DE EXPERIMENTOS	32
4.1 Cenário das Simulações	33
4.2 Modelo Fatorial 2 ^k r	33

4.3	Avaliação dos Fatores.....	37
5.	AVALIAÇÃO DE DESEMPENHO.....	39
5.1	Tamanho de <i>Buffer</i> variável.....	40
5.2	Intervalo de geração de mensagens variável.....	45
5.3	Número de cópias variável.....	48
6.	CONCLUSÕES E SUGESTÕES.....	53
6.1	Trabalhos Futuros.....	53
6.2	Publicação.....	54
7.	REFERÊNCIAS BIBLIOGRÁFICAS.....	55
	APÊNDICE A.....	58
	APÊNDICE B.....	62

LISTA DE ACRÔNIMOS

AODV	Ad Hoc On-Demand Distance Vector
DSR	Dynamic Source Routing
DTN	Delay and Disruption Tolerant Networks
EMHC	Estimated Handling Message Capacity
IDE	Integrated Development Environment
IoT	Internet of Things
MHC	Message Handling Capacity
MNCBSS	Motion and Counting Based-Spray and Seek routing algorithm
ONE	Opportunistic Network Environment
PRoPHET	Probabilistic Routing Protocol using History of Encounters and Transitivity
QoN-ASW	Adaptive Spray and Wait routing algorithm based on Quality of Node
SnW	Spray and Wait
SRAMSW	Social Relationship based Adaptive Multi-Spray-and-Wait routing algorithm
TTL	Time-To-Live

LISTA DE FIGURAS

Figura 1: Transmissão Epidêmica de mensagens.....	17
Figura 2: Transmissão de mensagens do PRoPHET.....	20
Figura 3: Tipos de Distribuição da Fase Spray.....	21
Figura 4: Algoritmo QoN-ASW.....	26
Figura 5: Algoritmo QoN++.....	31

LISTA DE GRÁFICOS

Gráfico 1: Sobrecarga em rede para tamanho de <i>Buffer</i> variável	41
Gráfico 2: Sobrecarga em rede para tamanho de <i>Buffer</i> variável (SnW, QoN-ASW e QoN++).....	42
Gráfico 3: Taxas de entrega para tamanho de <i>Buffer</i> variável	42
Gráfico 4: Atraso médio para tamanho de <i>Buffer</i> variável.....	43
Gráfico 5: Sobrecarga em rede para intervalo de geração de mensagens variável.....	46
Gráfico 6: Sobrecarga em rede para intervalo de geração variável (SnW, QoN-ASW e QoN++).....	46
Gráfico 7: Taxas de entrega para intervalo de geração variável.....	47
Gráfico 8: Atraso médio para intervalo de geração variável.....	47
Gráfico 9: Sobrecarga em rede para número de cópias variável.....	49
Gráfico 10: Sobrecarga em rede para número de cópias variável (SnW, QoN-ASW e QoN++).....	50
Gráfico 11: Taxas de entrega para número de cópias variável.....	50
Gráfico 12: Atraso médio para número de cópias variável.....	51

LISTA DE TABELAS

Tabela 1: Fatores das simulações	32
Tabela 2: Níveis mínimo e máximo dos fatores	34
Tabela 3: Variação de X_i	35
Tabela 4: Primeira avaliação de fatores	37
Tabela 5: Segunda avaliação de fatores	38
Tabela 6: Config. da simulação 5.1 - <i>Buffer</i> variável	41
Tabela 7: Config. da simulação 5.2 – Intervalo de geração variável	45
Tabela 8: Config. da simulação 5.3 – Número de cópias variável	49

1. INTRODUÇÃO

A evolução tecnológica dos sistemas embarcados permite o processamento e o compartilhamento automatizado de dados entre dispositivos eletrônicos conectados por rede. Um conceito associado às relações entre esses dispositivos é o da Internet das Coisas (*Internet of Things - IoT*).

A IoT mescla múltiplas tecnologias de comunicação e processamento em tempo real, sensoriamento, aprendizagem de máquina, automatização de processos e gerenciamento de recursos de rede, oferecendo produtos e soluções com diversas aplicações práticas.

Dentre as áreas de pesquisa e de desenvolvimento potencialmente beneficiadas, menciona-se: interconexão de redes de sensores sem fio [1][2]; identificação do comportamento social dos dispositivos em uma rede [3]; arquitetura de cidades inteligentes [4]; monitoramento médico remoto [5] e sua redução de gastos financeiros [6]; a IIoT (*Industrial Internet of Things*) [7] e o seu potencial para gerar trilhões de dólares em produtos internos brutos a nível mundial [8]; controle de irrigação e de fertilização para a agricultura [9]; controle de tráfego de veículos [10]; entre outras.

Muitos destes dispositivos inteligentes são móveis. Em alguns cenários, eles estão em constante movimentação, o que torna o estabelecimento de rotas fim-a-fim de protocolos como AODV [11] e DSR [12] ineficaz.

A conectividade intermitente de redes móveis é uma característica das Redes Tolerantes a Atrasos e Desconexões (*Delay and Disruption Tolerant Networks - DTN*). Nelas, é impossível definir rotas pré-estabelecidas entre os nós da rede. Uma das abordagens mais comuns para contornar a instabilidade das conexões é conhecida como "*store-carry-forward*", na qual os nós de uma rede armazenam temporariamente os dados recebidos na memória e os encaminham a outros nós na ocorrência de certos eventos. Os critérios de retransmissão variam entre algoritmos de roteamento.

Desde o início dos anos 2000, pesquisas vêm sendo realizadas com o intuito de criar algoritmos de roteamento em DTNs que possibilitem a elevação das taxas de entrega de dados no menor intervalo de tempo possível. Para controlar o consumo energético e o tráfego de rede, estas pesquisas também buscam reduzir o número

de transmissões de dados desnecessárias ou redundantes. Em outras palavras, deseja-se reduzir a sobrecarga em rede.

Alguns algoritmos já foram amplamente investigados, sendo tradicionais na literatura especializada. Dentre eles, pode-se citar o Epidêmico [13], o P_{Ro}PHET [14] e o Spray and Wait (SnW) [15]. Protocolos e algoritmos de roteamento mais recentes podem ser modificações ou extensões de ideias apresentadas em modelos mais antigos.

O novo algoritmo QoN-ASW [16] realiza uma fusão de conceitos apresentados pelo P_{Ro}PHET e pelo Spray and Wait. Esse algoritmo foi selecionado como uma referência de estudo por conciliar o uso de funções de probabilidade de contato entre nós, herdadas do P_{Ro}PHET, com o controle do número máximo de cópias de uma mensagem da rede, introduzido pelo SnW. Apesar dos ganhos em taxa de entrega de dados e da redução no atraso médio, ele apresenta uma sobrecarga em rede superior ao do Spray And Wait. Nesta dissertação, propõe-se o algoritmo QoN++, uma modificação do QoN-ASW responsável pela redução da sobrecarga em redes.

O cenário de teste utilizado para avaliar o desempenho do QoN++ segue o comportamento real de dispositivos geradores e transmissores de mensagens no evento Infocom06. Um arquivo de *trace* do Haggler Project [17] convertido para o formato *StandardEventsReader* do simulador ONE (*Opportunistic Network Environment*) [18], versão 1.4.1, registra o histórico de conexões entre os dispositivos. O simulador opera em conjunto com o IDE Eclipse 2018-12.

O ONE utiliza o histórico de conexões entre dispositivos para executar os algoritmos de roteamento em DTN. Desse modo, o simulador fornece informações como sobrecarga na rede, taxa de entrega de pacotes, latência média, entre outras. O desempenho do QoN++ foi comparado ao de modelos mais revisados na literatura [13-15] e ao do QoN-ASW [16].

1.1 Descrição do Problema

Em DTN, as conexões entre os nós são instáveis. O repasse constante de dados entre nós, a cada contato, consome energia de transmissores e receptores e contribui para o aumento exagerado de tráfego na rede. A entrega das mensagens aos nós de destino é incerta, devendo ocorrer antes dos seus tempos de expiração. Diante das dificuldades em garantir um elevado percentual de entrega em um tempo relativamente curto, surgem duas perguntas:

- É possível desenvolver um novo algoritmo de roteamento, ou melhorar algum modelo existente, de tal modo que a sobrecarga na rede seja reduzida ao mesmo tempo em que as taxas de entrega sejam mantidas ou elevadas?
- É possível detectar, tendo como única informação o histórico de encontros prévios, os melhores nós para repassar dados sem saturar a rede?

Tais perguntas são desafios abertos à comunidade científica.

1.2 Motivação

A determinação de rotas de menor custo de transmissão de dados entre nós de uma DTN não pode ser baseada em um conhecimento global da topologia da rede, pois se pressupõe que os nós estejam em constante movimentação e que eles não tenham ciência da posição dos outros nós da rede. No entanto, observa-se que a movimentação humana geralmente segue padrões de deslocamento ou de relacionamento entre elementos de uma rede [14].

Enviar dados em um cenário de incertezas implica transmissões que podem ser mal sucedidas em termos da não entrega de mensagens ou do gasto exagerado de energia e memória dos nós da rede. Motivada pela redução do desperdício de dados, tempo, energia, memória, processamento e tráfego de rede, a presente dissertação tem como objetivo apresentar uma proposta de algoritmo que reduza a sobrecarga na rede, manter ou elevar as taxas de entrega e analisar as modificações no atraso médio de entrega de mensagens.

1.3 Objetivos

1.3.1 Geral

Desenvolver um algoritmo que reduza a sobrecarga na rede em um cenário de mobilidade dos nós, onde o posicionamento exato de outros nós é incerto em decorrência da ausência de conhecimento global da topologia da rede.

1.3.2 Específicos

- Elaborar um resumo de algoritmos tradicionais e outros do estado da arte para roteamento em redes DTN;
- Escolher e codificar algoritmos de benchmark;
- Propor e implementar um novo algoritmo para redução de sobrecarga em redes DTN;
- Elaborar um projeto de experimentos;
- Avaliar o desempenho do algoritmo proposto;
- Verificar que a redução da sobrecarga na rede não compromete as taxas de entrega de dados, sendo estas mantidas ou elevadas;
- Analisar como o novo algoritmo modifica o atraso na entrega de mensagens.

2. TRABALHOS RELACIONADOS

2.1 Roteamento Epidêmico

De acordo com o modelo epidêmico [13], cada nó mantém um vetor sumário de mensagens contidas em seu *buffer*. Dois nós compartilham seus vetores quando estão fisicamente próximos.

Em seguida, cada nó verifica quais mensagens não estão presentes em seu *buffer* e que podem ser adquiridas a partir do contato com o outro nó, solicitando suas transmissões. Dessa forma, cada mensagem a ser transmitida terá uma cópia criada, sendo repassada para o nó receptor de acordo com a Figura 1.

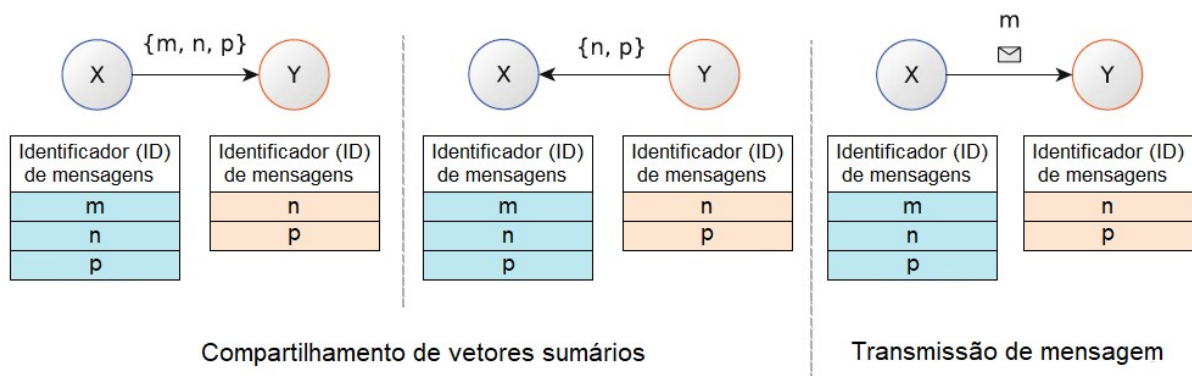


Figura 1 - Transmissão Epidêmica de mensagens. Os nós X e Y compartilham seus vetores entre si. Em seguida, o nó X transmite a Y a mensagem m que Y não possui. Adaptado de Misra, S. et.al. [19]

O objetivo do modelo epidêmico é maximizar as chances de entrega de mensagens e minimizar a latência. O procedimento realizado pelo modelo epidêmico é aconselhado para cenários ideais, sem restrição de memória ou de tempo de transmissão.

No entanto, dispositivos móveis reais (ex. *smartphones*) possuem recursos de armazenamento limitados e precisam gerenciar seus *buffers*. Quando um dispositivo receptor já está com a memória completamente utilizada, algumas mensagens podem ser deliberadamente descartadas de seu *buffer* para permitir a chegada de mensagens mais recentes. Outra possibilidade é que o dispositivo receptor opte por recusar as mensagens que o transmissor esteja tentando encaminhar, mantendo as mais antigas em seu *buffer*.

Enquanto as memórias dos dispositivos ainda possuem capacidade de armazenamento, as mensagens serão enviadas de um nó a outro. Conseqüentemente, há elevada sobrecarga na rede, acarretando taxas de entrega reduzidas e latências elevadas.

2.2 P_{Ro}PHET

Algoritmo orientado a probabilidades, baseado no histórico de encontros e de transitividade [14]. A estratégia do P_{Ro}PHET consiste em replicar e transmitir uma mensagem apenas em casos em que o nó receptor ainda não possua uma cópia desta e, ao mesmo tempo, apresente maior probabilidade de encontrar o nó de destino, em comparação ao nó transmissor.

As probabilidades de encontro são armazenadas em tabelas e atualizadas dinamicamente. Três funções gerenciam a sua funcionalidade:

2.2.1 Probabilidade de contato direto

$$P_{(a,b)} = P_{(a,b)old} + (1 - P_{(a,b)old}) * P_{init} \quad (1)$$

Aplicada quando os nós a e b estão a uma distância que permita interações diretas e troca de mensagens. $P_{(a,b)}$ e $P_{(a,b)old}$ correspondem às probabilidades atual e antiga de encontro entre a e b , respectivamente. O valor inicial de $P_{(a,b)old}$ é sempre igual a zero. P_{init} é uma constante de inicialização, definida por Lindgren et al. [14] como $P_{init} = 0.75$. Deve-se assegurar que $P_{(a,b)} \in [0,1]$.

Após o primeiro encontro, $P_{(a,b)}$ é igual a P_{init} , mas a probabilidade sofre um decréscimo natural com o passar do tempo. Nos encontros seguintes, o valor de $P_{(a,b)old}$ é o valor de $P_{(a,b)}$ previamente reduzido pela ação do envelhecimento, sendo utilizado na Eq. (1) para fornecer o novo $P_{(a,b)}$ após contato direto.

2.2.2 Probabilidade transitiva

$$P_{(a,c)} = P_{(a,c)old} + (1 - P_{(a,c)old}) * P_{(a,b)} * P_{(b,c)} * \beta \quad (2)$$

Utilizada nos casos em que um nó encaminha mensagens por meio de intermediários. Supondo que o nó a mantenha contato com o nó b e que este interaja repetidas vezes com um conjunto de nós c , é possível calcular ou atualizar a probabilidade de envio de uma mensagem de a para cada nó c por intermédio de b .

$P_{(a,c)}$ corresponde às probabilidades de contato entre a e os nós c , $P_{(a,b)}$ é igual à probabilidade de contato entre a e b e, de modo semelhante, $P_{(b,c)}$ é a probabilidade de contato entre b e os nós c . O parâmetro $\beta \in [0,1]$ controla o nível de influência da transitividade no cálculo da probabilidade, definido em [14] como $\beta = 0.25$.

A cada contato entre os nós a e b , o nó a atualiza a probabilidade de contato com cada nó c que já tenha uma probabilidade de contato com o nó b . Isso é realizado através do produto $P_{(a,b)} * P_{(b,c)} * \beta$, o qual está associado à probabilidade antiga $P_{(a,c)old}$ para a obtenção da nova $P_{(a,c)}$.

2.2.3 Fator de envelhecimento

$$P_{(a,b)} = P_{(a,b)old} * \xi^\tau \quad (3)$$

Quando dois nós a e b permanecem desconectados por algum tempo, o valor da probabilidade $P_{(a,b)}$ sofre uma redução natural. O parâmetro $\xi \in [0,1)$ é a constante de envelhecimento e τ representa o número de unidades de tempo transcorridas após a última atualização de $P_{(a,b)}$. O valor especificado para a constante é $\xi = 0.98$ [14].

Lindgren et al. [14] definiram empiricamente os valores fixos para P_{init} , β e ξ após um conjunto inicial de experimentos indicar que tais valores são escolhas adequadas para os parâmetros.

De acordo com o roteamento PProPHET, os nós a e b compartilham seus vetores de mensagens e suas tabelas de probabilidades (TP), atualizadas em cada nó. De acordo com os valores tabelados, os nós (a,b) podem criar cópias das mensagens e transmiti-las ou podem apenas se limitar a manter as mensagens em seus *buffers*, sem criar cópias para envios. A Figura 2 ilustra o processo.

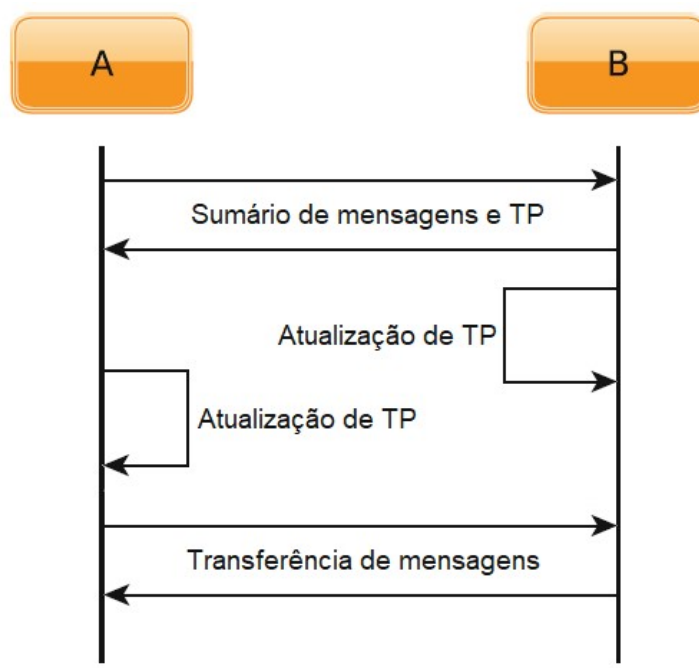


Figura 2 - Transmissão de mensagens do PProPHET. Adaptado de Misra, S. et.al. [19]

Uma característica indesejável do PProPHET é a sua incapacidade em representar certas dinâmicas do mundo real. Por exemplo, em cenários de mobilidade, conexões entre nós podem ser interrompidas diversas vezes em curtos intervalos de tempo, sendo interpretadas pelo PProPHET como novos encontros. Tal comportamento não corresponde a padrões de movimentação humana no decorrer de horas ou dias.

2.3 Spray and Wait

Também conhecido como *SnW* [15], este algoritmo apresenta uma vantagem em relação aos antecessores: o número máximo de cópias de uma mensagem é limitado, evitando a sobrecarga em DTN que pode ser percebida em modelos mais antigos.

O *Spray and Wait* recebe este nome por causa da forma como é estruturado. Ele é dividido em duas etapas: Fase *Spray* e Fase *Wait*.

2.3.1 Fase *Spray*

Considera-se que um nó possua uma mensagem para transmissão, cujo número máximo de cópias que possam circular na rede seja igual a N . O nó já pode transportar cópias desta mensagem prontas em sua memória ou pode criá-las

apenas durante um contato, para economizar buffer. Este nó, ao entrar em contato com outros nós da rede, pode repassar-lhes algumas cópias, caso estes ainda não as possuam. Há dois modelos básicos de compartilhamento: *source* e binário.

De acordo com o modelo *source*, o nó transmissor fornece uma cópia da mensagem a outro nó, mantendo as demais cópias para si. Ao encontrar um novo contato, ele repete o processo. Dessa forma, os primeiros $N-1$ nós a interagir com portadores da mensagem receberão uma cópia cada.

De acordo com o modelo binário, o número de cópias da mensagem que o transmissor fornece a um receptor é dado pela função piso da metade de N . Em outras palavras, o transmissor repassa $\lfloor N/2 \rfloor$ cópias da mensagem ao receptor, retendo o valor complementar $\lceil N/2 \rceil$ para si. Ambos os nós (transmissor e receptor) podem repetir o processo, dividindo e repassando as mensagens a outros contatos encontrados.

A Figura 3 mostra os diferentes tipos de distribuição de cópias de mensagens. Quando o nó tem apenas uma cópia restante, ele encerra o modo *Spray* e prossegue à fase *Wait*, explicada na seção 2.3.2. Caso o nó tenha conseguido entregar a mensagem durante a fase *Spray*, a operação do modelo é finalizada.

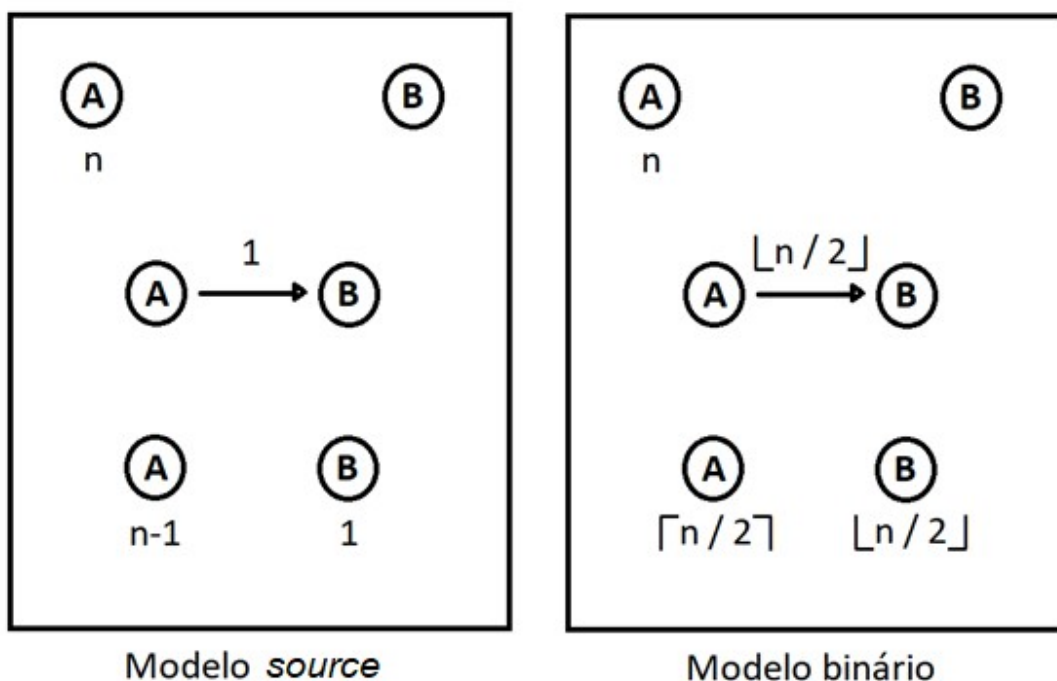


Figura 3 - Tipos de Distribuição da Fase *Spray*. No modelo de distribuição *source*, o nó A possui N cópias de uma mensagem e encaminha uma cópia ao nó B. No modelo binário, o nó A encaminha $\lfloor N/2 \rfloor$ cópias e mantém $\lceil N/2 \rceil$ para si.

2.3.2 Fase *Wait*

Quando um nó portador de uma mensagem contém apenas uma única cópia para repasse, diz-se que ele encerrou a fase *Spray* e ingressou na fase *Wait*. Nesta fase, o nó portador deverá realizar uma transmissão direta ao nó de destino, caso haja enlace estabelecido entre eles. Se não houver enlace, o portador deverá aguardar até a ocorrência de uma conexão com o destino. Após a entrega da mensagem, o nó de destino descartará futuras entregas da mesma mensagem.

No entanto, se tais conexões *jamaiz* ocorrerem, o nó portador poderá armazenar a mensagem em seu *buffer* indefinidamente ou durante um intervalo de tempo finito, até que a mensagem seja descartada sem ser entregue ao destino. Deste modo, o comportamento dos nós durante a fase *Wait* tende a ser pouco eficiente.

Para reverter essa limitação, devem-se criar condições de repasse para mensagens na fase *Wait*. O algoritmo *Adaptive Spray and Wait Based on Quality of Nodes in DTN* (QoN-ASW) [16] faz isso.

2.4 QoN-ASW: Adaptive Spray and Wait Based on Quality of Nodes in DTN

À semelhança do *Spray and Wait*, o nó portador de uma mensagem possui uma determinada quantidade de cópias em seu *buffer*. Quando este nó está em condições de realizar uma transferência, um valor de *Qualidade do Nó* é calculado e indica quantas cópias serão enviadas e quantas serão retidas [16].

Consideram-se dois nó i e j , conectados por um enlace. Para realizar o cálculo da qualidade do nó i , o QoN-ASW considera duas funções: capacidade de manipulação de mensagens (U_{MHC}) e probabilidade de entrega (U_P). Estas funções são calculadas separadamente e ponderadas em uma função unificada que determina a Qualidade do Nó.

2.4.1 Capacidade de manipulação de mensagens

O tempo total de simulação é dividido em uma determinada quantidade de *slots*, de mesma largura. Os *slots* são enumerados sequencialmente no decorrer da

simulação e representados pela variável k . Por exemplo, o primeiro *slot* é representado por $k = 1$, enquanto o quinto *slot* é representado por $k = 5$.

A capacidade estimada do nó i em um dado *slot* k ($EMHC_i^k$) [16] é o número estimado de cópias de mensagens que o nó i irá repassar durante o *slot* k .

$$EMHC_i^k = \begin{cases} \frac{m_{cur}}{\Delta t} * TS, & se\ k = 1 \\ \frac{m_{cur} - \sum_{x=1}^{k-1} m_x}{\Delta t} * TS, & se\ k > 1 \end{cases} \quad (4)$$

De acordo com a Eq. (4), m_{cur} representa o número de cópias de mensagens que o nó i encaminhou até o instante de cálculo. O valor de m_x representa o número de cópias encaminhadas em um dado *slot* x anterior ao *slot* atual de cálculo. Por exemplo, se o instante atual estiver inserido no *slot* $k = 3$, então m_1 e m_2 representam o número de cópias encaminhadas durante os *slots* 1 e 2, respectivamente. Portanto, o somatório $\sum_{x=1}^{k-1} m_x$ representa o número de cópias encaminhadas durante toda a simulação, do *slot* 1 até o *slot* imediatamente anterior ao atual ($k-1$).

A variável Δt denota o tempo transcorrido desde o fim do último *slot* e TS é a largura de um *slot*. Consideramos o mesmo número de *slots* adotados por Cui et al. [16], para fins de comparação/validação: $k = 40$ *slots*, com $TS = 1$ hora.

A capacidade de manipulação de mensagens é o número ponderado de cópias de mensagens que o nó i repassou até o instante de cálculo, situado no *slot* de tempo k . A capacidade de manipulação é representada por MHC_i^k e sua principal utilidade é informar aos nós da rede a habilidade que um nó tem para encaminhar mensagens.

$$MHC_i^k = \begin{cases} EMHC_i^k, & se\ k = 1 \\ \alpha * MHC_i^{k-1} + (1 - \alpha) * EMHC_i^k, & se\ k > 1 \end{cases} \quad (5)$$

O cálculo de MHC_i^k depende do instante de tempo em que é realizado. Se ele ocorrer no primeiro *slot*, o valor de MHC_i^k é igual a $EMHC_i^k$. A partir do segundo *slot*, MHC_i^k é uma ponderação entre a capacidade do *slot* anterior (MHC_i^{k-1}) e a capacidade estimada do *slot* atual ($EMHC_i^k$).

A ponderação é controlada pelo parâmetro de amortecimento $\alpha \in [0,1]$. Na prática, α define como a capacidade MHC_i^{k-1} do *slot* anterior interfere na capacidade do *slot* atual MHC_i^k . O valor de amortecimento é o mesmo adotado por Cui et al. [16] e Wang et al. [20], para fins de validação: $\alpha = 0.15$.

A função U_{MHC} denota a utilidade relativa do nó i para entregar a mensagem ao destino, comparado ao nó j

$$U_{MHC} = \frac{MHC_i^k}{MHC_i^k + MHC_j^k} \quad (6)$$

2.4.2 Probabilidade de Entrega

A Eq. (7) apresenta a Força de Conectividade (FC) entre os nós a e b .

$$FC_{(a,b)} = \frac{\sum_{l=1}^n (S_{ab}^l - E_{ab}^l)}{T} \quad (7)$$

De acordo com a Eq.(7), n indica o número de conexões entre a e b , no decorrer do tempo. Para cada conexão individual l , são registrados os respectivos instantes de início (S_{ab}^l) e de fim (E_{ab}^l). A diferença entre ambos equivale à duração da conexão l entre a e b . O somatório destas diferenças representa o tempo em que a e b estiveram conectados durante a simulação, cujo tempo total é dado por T .

O QoN-ASW utiliza funções de probabilidade semelhantes às do PProPHET, com atualizações por contato direto, por transitividade e por envelhecimento. Apenas a função de contato direto (Eq. (1)) apresenta modificações na fórmula:

$$P_{(a,b)} = P_{(a,b)old} + (1 - P_{(a,b)old}) * P_{int} * \lambda^{FC_{(a,b)}} \quad (8)$$

As demais funções do PProPHET, representadas pelas Eq. (2) e (3), são mantidas. Na Eq. (8), $FC_{(a,b)}$ é utilizada em conjunto com o parâmetro λ , o qual corresponde à influência de FC na probabilidade do nó a entregar uma cópia de mensagem ao nó b .

O algoritmo QoN-ASW modifica os valores dos parâmetros das expressões (2), (3) e (8). Enquanto o PProPHET trabalha com $\beta = 0.25$, $\xi = 0.98$ e $P_{init} = 0.75$ [14], Cui et al. [16] descobriram que os valores otimizados para os parâmetros são

$\beta = 0.1$, $\xi = 0.8$ e $P_{init} = 0.7$, os quais permitem taxas de entrega mais elevadas, assim como redução do atraso médio e da sobrecarga em rede.

A função utilidade da probabilidade de entrega, U_P , é uma relação entre a probabilidade de entrega do nó i ao destino d e a soma das probabilidades de entrega dos nós i e j ao referido destino.

$$U_P = \frac{P_{(i,d)}}{P_{(i,d)} + P_{(j,d)}} \quad (9)$$

2.4.3 Qualidade dos Nós

As Eq. (6) e (9) fornecem a base de cálculo da Qualidade do Nó.

$$QoN_i = \eta * U_{MHC} + (1 - \eta) * U_P \quad (10)$$

Onde η é um fator de ajuste de importância entre as duas funções. De acordo com [16], $\eta = 0.7$. No entanto, tal valor pode ser modificado caso haja interesse em dar maior ou menor ênfase à capacidade de transmissão de mensagens ou às probabilidades de contato.

2.4.4 O Algoritmo do QoN-ASW

Trata-se de uma adaptação do *Spray and Wait*, em que os valores de Qualidade dos Nós são utilizados para determinar a quantidade de cópias que serão repassadas na Fase *Spray*, enquanto a probabilidade é utilizada para determinar o repasse ou retenção de uma cópia durante a Fase *Wait*.

Algoritmo 1 - QoN-ASW

- 1: **se** o nó v_i encontrar o nó v_j
- 2: atualize MHC_i^k e MHC_j^k
- 3: atualize $P_{(i,j)}$
- 4: **para** $\forall m_l \in v_i$ e $m_l \notin v_j$
- 5: $L_{i_{old}}(m_l) =$ número de cópias de m_l
- 6: $v_d =$ nó de destino de m_l
- 7: **se** ($v_d == v_j$)
- 8: v_i encaminha mensagem m_l a v_j
- 9: v_i remove mensagem m_l do buffer de v_i

```

10:      senão se ( $L_{i_{old}}(m_l) > 1$ )
11:          calcule  $QoN_i$  e  $QoN_j$ 
12:           $L_{i_{new}}(m_l) = \lfloor QoN_i \cdot L_{i_{old}}(m_l) \rfloor$ 
13:           $L_{j_{new}}(m_l) = L_{i_{old}}(m_l) - L_{i_{new}}(m_l)$ 
14:          se ( $L_{j_{new}}(m_l) > 0$ )
15:               $v_i$  encaminha  $L_{j_{new}}(m_l)$  cópias de  $m_l$  a  $v_j$ 
16:               $v_i$  mantém  $L_{i_{new}}(m_l)$  cópias de  $m_l$ 
17:          fim se
18:      senão se ( $L_{i_{old}}(m_l) == 1$ )
19:          se ( $P_{(j,d)} > P_{(i,d)}$ )
20:               $v_i$  encaminha mensagem  $m_l$  a  $v_j$ 
21:               $v_i$  remove mensagem  $m_l$  de seu buffer
22:          fim se
23:      fim se
24:  fim para
25: fim se

```

Figura 4 - Algoritmo QoN-ASW. Traduzido da versão original de CUI, J. et.al. [16]

O nó transmissor de uma ou várias cópias da mensagem é chamado v_i , enquanto o potencial receptor é denominado v_j e o vértice de destino, v_d . O instante de tempo em que os cálculos são realizados está situado no *slot* k .

A primeira etapa do algoritmo está descrita nas linhas 1-3. Se v_i e v_j estiverem em contato, ambos atualizam as suas capacidades, a probabilidade de contato direto entre eles e as probabilidades transitivas.

Então, para cada mensagem l contida no *buffer* de v_i e que *não* estiver contida no *buffer* de v_j , atribui-se a $L_{i_{old}}(m_l)$ o número inicial de cópias de l (linhas 4-5).

Caso haja conexão direta entre v_i e v_d , a mensagem é entregue ao destino e removida do *buffer* de v_i (linhas 7-9).

Em outros casos, deve-se verificar se o número de cópias de l é igual ou superior a um. Se for maior que um, procede-se à fase *Spray* (linhas 10-17). Se for igual, procede-se à fase *Wait* (linhas 18-22).

A fase *Spray* exige o cálculo do QoN de v_i . O número de cópias de l retidas no *buffer* de v_i após o cálculo de QoN é dado por $L_{i_{new}}(m_l)$. A diferença entre os valores inicial e final de cópias no *buffer* de v_i corresponde ao número de cópias para transferência ao *buffer* de v_j , sendo registrado na variável $L_{j_{new}}(m_l)$. Se $L_{j_{new}}(m_l)$ for

maior que zero, realiza-se o compartilhamento de cópias da mensagem l e o *buffer* de v_i é atualizado.

A fase *Wait* baseia-se na comparação de probabilidades entre v_i e v_j . Caso $P_{(j,d)} > P_{(i,d)}$, o nó v_i passa sua cópia de l para v_j . Caso contrário, v_i a retém para si.

Em relação aos algoritmos anteriores, o QoN-ASW oferece taxas de entrega mais elevadas e latências menores. Isso pode ser explicado pelo número máximo limitado de cópias por mensagens, pela distribuição de cópias baseada em uma medida matemática de qualidade dos nós e pela habilidade de transmitir uma mensagem na fase *Wait*, evitando o comportamento passivo típico do SnW.

2.5 Outros Trabalhos

Outras pesquisas fazem uso do número máximo de cópias ou de qualidades sociais dos nós, que estão relacionadas à capacidade de encaminhar mensagens ou de renovar listas de contatos.

Du et al. apresentam [21] *Motion and Counting Based-Spray and Seek routing algorithm* (MNCBSS). Trata-se de uma adaptação do *Spray and Wait*, denominada *Spray and Seek*, dividida em duas fases.

Na fase *Spray*, consideram-se as direções dos movimentos de dois nós conectados por enlace e o número de cópias da mensagem do nó transmissor. Se as direções forem muito diferentes e o número de cópias alcançar um dado *threshold*, metade das cópias são compartilhadas. Se apenas uma das condições for satisfeita, somente uma cópia é repassada.

Na fase *Seek*, um contador de encontros influencia o Parâmetro Ativo do nó, o qual representa de modo mais apurado e amortecido a atividade social deste. Então, os dois nós compartilham seus Parâmetros e o mais ativo ficará com a cópia da mensagem.

Guan et al. descrevem o *Social Relationship based Adaptive Multi-Spray-and-Wait routing algorithm* (SRAMSW) [22]. Este modelo de roteamento determina que os vizinhos inseridos na atual lista de contatos de um nó devem ser ordenados em uma lista de prioridades. Para tal ordenação, utiliza-se uma métrica baseada em conceitos como Similaridade, Centralidade e Amizade. À parte, as mensagens armazenadas no *buffer* são associadas a tempos de permanência, que não podem

extrapolar um *timeout*. Para cada mensagem no *buffer* de um nó, deve-se verificar se ela extrapolou o *timeout*. Se este limite for ultrapassado, as mensagens devem ser encaminhadas a N vizinhos com as maiores métricas na lista de prioridades. Caso contrário, as mensagens devem ser enviadas a N vizinhos com maiores similaridades em relação ao destino.

Wang et al propõem [23] um roteamento baseado em intensidade e estabilidade de conexões para avaliar a capacidade de conexão de um nó. A intensidade é medida a partir do tempo médio de separação e a estabilidade equivale à raiz quadrada deste tempo. No entanto, se as mensagens forem entregues a nós com boa capacidade de conexão, mas com pouca habilidade de repasse, as taxas de entrega podem apresentar queda.

3. ALGORITMO QoN++

De acordo com os algoritmos mencionados, os nós da rede mantêm cópias de mensagens em seus *buffers* e utilizam diferentes estratégias para localizar os nós de destino e entregar-lhes os pacotes. Quando um nó portador de uma mensagem a encaminha diretamente ao nó de destino, apenas este par de nós fica ciente da entrega. Os demais nós da rede que possuem cópias desta mensagem não são notificados, podendo mantê-las em seus *buffers* até que o TTL delas expire. O TTL (*Time-to-Live*) de uma mensagem significa o tempo que uma mensagem pode circular na rede antes de ser descartada ou revalidada.

Desse modo, os nós que não foram informados sobre tais entregas persistem na tarefa de localizar nós de destino ou outros nós que possuam maiores chances de realizar a entrega final. Trata-se de uma atividade desnecessária, que contribui para a elevação da sobrecarga na rede.

Na presente pesquisa, introduzimos o novo QoN++, que realiza modificações no QoN-ASW através de duas características adicionais:

A primeira consiste em tratar a entrega de mensagens como eventos que devem ser notificados a todos os nós da rede. Se um nó for informado sobre a entrega de uma mensagem e se ele detectar cópias desta mensagem em seu *buffer*, as cópias serão removidas do *buffer*. Isso é realizado através do compartilhamento de tabelas de entregas entre contatos, sendo continuamente atualizadas a cada encontro.

A segunda característica é o ajuste da estratégia de encaminhamento das cópias de mensagens em função do valor de QoN, que está situado em um intervalo $[0,1]$. Um nó com qualidade muito baixa em relação a seus contatos pode não ser a melhor escolha para encaminhar mensagens até o destino final. No entanto, ele pode ser útil como intermediário para futuros encaminhamentos a nós com qualidades mais elevadas. Para essa finalidade, ele não necessita dispor de um grande número de cópias e o cálculo do repasse segue um padrão fixo. Desse modo, o valor numérico de QoN será aplicado no cálculo do número de cópias para repasse apenas nos casos em que este valor não seja muito próximo do valor mínimo (zero) ou máximo (um). Após uma série de experimentos, foi delimitado um intervalo médio $[0.3, 0.6]$ para valores de QoN, em que eles são usados no cálculo do número de cópias que devem ser repassadas ou mantidas nos *buffers* dos nós.

Foram analisados outros intervalos, mas os valores obtidos em [0.3, 0.6) ofereceram maiores taxas de entrega, menor sobrecarga em rede e poucas diferenças nos atrasos médios.

O Algoritmo 2 ilustra o QoN++. O nó v_i mantém a tabela $F(i)$ com todas as mensagens cujas entregas finais já lhe tenham sido informadas. Se v_i já teve contato prévio com um dado nó v_j , v_i precisa saber se houve alterações em $F(j)$. Portanto, chamamos de $F(i,j)_{new}$ as informações mais recentes de $F(j)$, que estão sendo fornecidas de v_j para v_i . $F(i,j)_{old}$ são informações antigas de $F(j)$, a qual v_i teve acesso em algum momento prévio. Caso v_i e v_j estejam se encontrando pela primeira vez, $F(i,j)_{old}$ é nula.

Algoritmo 2 – QoN++

```

01: se o nó  $v_i$  encontrar o nó  $v_j$ 
02:   se  $(F(i,j)_{old} == \text{nula})$  ou  $(F(i,j)_{new} != F(i,j)_{old})$ 
03:     para  $\forall m_l \in F(j)$  e  $m_l \notin F(i)$ 
04:        $v_i$  adiciona  $m_l$  a  $F(i)$ 
05:       se  $(m_l \in v_i)$ 
06:          $v_i$  remove  $m_l$  de seu buffer
07:       fim se
08:     fim para
09:   fim se
10:   atualize  $MHC_i^k$  e  $MHC_j^k$ 
11:   atualize  $P_{(i,j)}$ 
12:   para  $\forall m_l \in v_i$  e  $m_l \notin v_j$ 
13:      $L_{i_{old}}(m_l) =$  número de cópias de  $m_l$ 
14:      $v_d =$  nó de destino de  $m_l$ 
15:     se  $(v_d == v_j)$ 
16:        $v_i$  encaminha  $m_l$  a  $v_j$ 
17:        $v_i$  adiciona  $m_l$  a  $F(i)$ 
18:        $v_i$  remove  $m_l$  de seu buffer
19:     senão se  $(L_{i_{old}}(m_l) > 1)$ 
20:       calcule  $QoN_i$ 
21:       se  $(QoN_i < 0,3)$ 
22:          $L_{i_{new}}(m_l) = 1$ 
23:          $L_{j_{new}}(m_l) = L_{i_{old}}(m_l) - 1$ 
24:       senão se  $(QoN_i \geq 0,3)$  e  $(QoN_i < 0,6)$ 
25:          $L_{i_{new}}(m_l) = \lfloor QoN_i \cdot L_{i_{old}}(m_l) \rfloor$ 
26:          $L_{j_{new}}(m_l) = L_{i_{old}}(m_l) - L_{i_{new}}(m_l)$ 

```

```

27:      senão
28:           $L_{i_{new}}(m_l) = L_{i_{old}}(m_l) - 1$ 
29:           $L_{j_{new}}(m_l) = 1$ 
30:      fim se
31:      se ( $L_{j_{new}}(m_l) > 0$ )
32:           $v_i$  encaminha  $L_{j_{new}}$  cópias de  $m_l$  a  $v_j$ 
33:           $v_i$  mantém  $L_{i_{new}}$  cópias de  $m_l$ 
34:      fim se
35:      senão se ( $L_{i_{old}}(m_l) == 1$ )
36:          se ( $P_{(j,d)} > P_{(i,d)}$ )
37:               $v_i$  encaminha  $m_l$  a  $v_j$ 
38:               $v_i$  remove  $m_l$  de seu buffer
39:          fim se
40:      fim se
41:  fim para
42: fim se

```

Figura 5 - Algoritmo QoN++

A atualização do *buffer* (linhas 2-6) precede os encaminhamentos de dados. Quando dois nós se encontram, eles compartilham uma lista de todas as mensagens cujas entregas finais já lhes foram informadas. Esse conhecimento é transmitido em cadeia por toda a rede. Em seguida, os nós atualizam suas capacidades, as probabilidades de contato direto e as probabilidades transitivas (linhas 10-11). Depois, para cada mensagem ainda presente em seus *buffers*, os nós analisam os destinos de cada mensagem e o respectivo número de cópias. (linhas 12-14).

A transmissão pode ser realizada diretamente ao nó de destino (linhas 15-18), o que acarretaria atualização da tabela $F(i)$. Caso contrário, analisa-se o número de cópias da mensagem. Se for superior a um (linhas 19-34), o nó transmissor ingressa na fase *Spray* e pode: i) reter apenas uma única cópia da mensagem caso sua QoN for baixa (linhas 21-23); ii) recorrer a uma divisão balanceada pela QoN se sua qualidade for média (linhas 24-26); ou iii) repassar apenas uma cópia (linhas 27-29) no caso de uma QoN elevada. Definida a estratégia de alocação de cópias, o encaminhamento é realizado (linhas 31-34). No caso de uma única cópia da mensagem (linhas 35-39), o nó transmissor entra na fase *Wait* e o encaminhamento é similar ao QoN-ASW.

4. PROJETO DE EXPERIMENTOS

O principal objetivo da presente pesquisa é apresentar um algoritmo, denominado QoN++, capaz de reduzir a sobrecarga em rede em relação ao desempenho de QoN-ASW. A fórmula da sobrecarga é dada pela Eq. 11.

$$Sobrecarga = \frac{N_{rep} - N_{ent}}{N_{ent}} \quad (11)$$

O numerador desta expressão é a diferença entre o número de mensagens repassadas entre os nós (N_{rep}) e de mensagens entregues ao destino (N_{ent}).

A sobrecarga indica a necessidade de retransmitir diversas vezes a mesma mensagem até a sua entrega. Essa retransmissão não se restringe apenas ao nó criador da mensagem. Qualquer outro nó que possua uma ou mais cópias desta mensagem e que necessite retransmiti-la causa um aumento no número de encaminhamentos na rede. A redução da sobrecarga diminui o tráfego de informações redundantes na rede e ajuda os nós a economizar energia.

Para realizar um projeto de experimentos que evidencie os ganhos de QoN++, é necessário criar um cenário para as simulações, escolher um modelo de projeto adequado e definir um conjunto de fatores para investigação.

Fatores são as variáveis de entrada de um experimento, enquanto *níveis* são valores específicos atribuídos a cada um dos fatores [24]. Os fatores escolhidos para as simulações são exibidos na Tabela 1, acompanhados por uma letra de identificação:

Tabela 1: Fatores das simulações

Identificação	Fator (unidade de medida)
A	Time-to-Live (min)
B	Tamanho do <i>buffer</i> (MBytes)
C	Tempo de simulação (h)
D	Intervalo de geração (seg)
E	Número de cópias

Este capítulo fornece detalhes sobre o cenário das simulações e aborda o modelo Fatorial $2^k r$ [24], escolhido para os experimentos. Verificar-se-á que a escolha do modelo possui duas importantes vantagens: redução do número de experimentos e, ao mesmo tempo, oferecer informações sobre como os fatores interagem entre si e interferem nos valores finais da variável resposta, que é a sobrecarga. Por fim, os cinco fatores escolhidos serão avaliados e os mais relevantes serão testados com mais níveis no próximo capítulo.

4.1 Cenário das Simulações

O cenário é o evento IEEE Infocom 2006 [17], ocorrido em Barcelona, Espanha, de 23 a 29 de Abril de 2006. Durante esta conferência, 20 dispositivos estacionários e 78 dispositivos móveis enviavam mensagens. Os dispositivos móveis estavam em posse de estudantes de diversas comunidades, as quais eram caracterizadas por diferentes locais de origem, tópicos de pesquisa e áreas de interesse.

O histórico de conexões entre esses dispositivos está descrito em um arquivo de *trace* registrado pelo Huggle Project [17], adaptado ao formato *StandardEventsReader* e reconhecido pelo simulador ONE - *Opportunistic Network Environment* [18], versão 1.4.1, utilizado como ambiente de programação e de testes.

O simulador é executado em conjunto com o IDE Eclipse 2018-12, no Sistema Operacional Windows 10 Home Single Language x64. O computador utilizado nos testes dispõe de 8GB de memória RAM e um processador Intel Core i5-8265U de 1.6GHz.

4.2 Modelo Fatorial $2^k r$

O modelo de projeto Fatorial $2^k r$ é frequentemente usado em experimentos onde é necessário estudar o efeito das interações de múltiplos fatores na resposta, sendo especialmente útil nos primeiros estágios de investigação. Cada fator assume apenas dois níveis – um máximo e um mínimo – e essa característica resulta na menor quantidade necessária de testes para a obtenção de uma análise fatorial completa [24]. Os níveis atribuídos a cada fator da simulação são fornecidos pela Tabela 2.

Tabela 2: Níveis mínimo e máximo dos fatores

Fator	Valor mínimo	Valor máximo
A	100 min	800 min
B	5 MB	40 MB
C	25h	60h
D	30 seg	100 seg
E	2 cópias	30 cópias

Os níveis mínimo e máximo podem ser atribuídos a variáveis codificadas X_i . Por exemplo, $X_A = -1$ se TTL = 100 minutos ou $X_A = 1$ se TTL = 800 minutos. Tais variáveis codificadas são utilizadas em um modelo de regressão não linear construído para representar os efeitos dos fatores e de todas suas possíveis interações na resposta, dado pela Eq. 12. O erro experimental e também é estimado.

$$\bar{y} = q_0 + q_A X_A + q_B X_B + \dots + q_{BCDE} X_B X_C X_D X_E + q_{ABCDE} X_A X_B X_C X_D X_E + e \quad (12)$$

Os coeficientes q_i são utilizados nos cálculos das porções de variação da resposta atribuídas a cada fator ou a cada interação entre fatores. Cada experimento é modelado de acordo com a Eq. 12. Como há cinco fatores neste projeto e desejamos estimar o erro, o modelo é descrito em um conjunto de 32 experimentos que devem ser repetidos r vezes. Nesta pesquisa, foram adotadas $r = 30$ repetições para cada experimento. A média das respostas obtidas é dada pela variável \bar{y} . Os valores de $X = \{X_A, X_B, X_C, X_D, X_E\}$ variam de acordo com a Tabela 3, determinando os sinais dos coeficientes q_i .

As equações completas para cada experimento são fornecidas no Apêndice A. Seguem alguns exemplos:

$$\bar{y}_1 = q_0 - q_A - q_B - q_C - q_D - q_E + \dots + q_{ABDE} + q_{ACDE} + q_{BCDE} - q_{ABCDE} + e$$

$$\bar{y}_6 = q_0 - q_A - q_B + q_C - q_D + q_E + \dots - q_{ABDE} + q_{ACDE} + q_{BCDE} - q_{ABCDE} + e$$

$$\bar{y}_{11} = q_0 - q_A + q_B - q_C + q_D - q_E - \dots + q_{ABDE} - q_{ACDE} + q_{BCDE} - q_{ABCDE} + e$$

$$\bar{y}_{16} = q_0 - q_A + q_B + q_C + q_D + q_E - \dots - q_{ABDE} - q_{ACDE} + q_{BCDE} - q_{ABCDE} + e$$

$$\overline{y_{17}} = q_0 + q_A - q_B - q_C - q_D - q_E - \dots - q_{ABDE} - q_{ACDE} + q_{BCDE} + q_{ABCDE} + e$$

$$\overline{y_{22}} = q_0 + q_A - q_B + q_C - q_D + q_E - \dots + q_{ABDE} - q_{ACDE} + q_{BCDE} + q_{ABCDE} + e$$

$$\overline{y_{27}} = q_0 + q_A + q_B - q_C + q_D - q_E + \dots - q_{ABDE} + q_{ACDE} + q_{BCDE} + q_{ABCDE} + e$$

$$\overline{y_{32}} = q_0 + q_A + q_B + q_C + q_D + q_E + \dots + q_{ABDE} + q_{ACDE} + q_{BCDE} + q_{ABCDE} + e$$

Tabela 3: Variação de X_i

Exp	X_A	X_B	X_C	X_D	X_E
1	-1	-1	-1	-1	-1
2	-1	-1	-1	-1	1
3	-1	-1	-1	1	-1
4	-1	-1	-1	1	1
5	-1	-1	1	-1	-1
6	-1	-1	1	-1	1
7	-1	-1	1	1	-1
8	-1	-1	1	1	1
9	-1	1	-1	-1	-1
10	-1	1	-1	-1	1
11	-1	1	-1	1	-1
12	-1	1	-1	1	1
13	-1	1	1	-1	-1
14	-1	1	1	-1	1
15	-1	1	1	1	-1
16	-1	1	1	1	1
17	1	-1	-1	-1	-1
18	1	-1	-1	-1	1
19	1	-1	-1	1	-1
20	1	-1	-1	1	1
21	1	-1	1	-1	-1
22	1	-1	1	-1	1
23	1	-1	1	1	-1
24	1	-1	1	1	1
25	1	1	-1	-1	-1
26	1	1	-1	-1	1
27	1	1	-1	1	-1
28	1	1	-1	1	1
29	1	1	1	-1	-1
30	1	1	1	-1	1
31	1	1	1	1	-1
32	1	1	1	1	1

Resolve-se o sistema de equações para calcular os coeficientes em função das respostas. As equações completas estão no Apêndice B. Alguns exemplos:

$$\begin{aligned}
 q_0 &= \frac{1}{32} (\bar{y}_1 + \bar{y}_2 + \bar{y}_3 + \bar{y}_4 + \bar{y}_5 + \bar{y}_6 + \dots + \bar{y}_{27} + \bar{y}_{28} + \bar{y}_{29} + \bar{y}_{30} + \bar{y}_{31} + \bar{y}_{32}) \\
 q_A &= \frac{1}{32} (-\bar{y}_1 - \bar{y}_2 - \bar{y}_3 - \bar{y}_4 - \bar{y}_5 - \bar{y}_6 - \dots + \bar{y}_{27} + \bar{y}_{28} + \bar{y}_{29} + \bar{y}_{30} + \bar{y}_{31} + \bar{y}_{32}) \\
 q_B &= \frac{1}{32} (-\bar{y}_1 - \bar{y}_2 - \bar{y}_3 - \bar{y}_4 - \bar{y}_5 - \bar{y}_6 - \dots + \bar{y}_{27} + \bar{y}_{28} + \bar{y}_{29} + \bar{y}_{30} + \bar{y}_{31} + \bar{y}_{32}) \\
 q_C &= \frac{1}{32} (-\bar{y}_1 - \bar{y}_2 - \bar{y}_3 - \bar{y}_4 + \bar{y}_5 + \bar{y}_6 + \dots - \bar{y}_{27} - \bar{y}_{28} + \bar{y}_{29} + \bar{y}_{30} + \bar{y}_{31} + \bar{y}_{32}) \\
 q_D &= \frac{1}{32} (-\bar{y}_1 - \bar{y}_2 + \bar{y}_3 + \bar{y}_4 - \bar{y}_5 - \bar{y}_6 + \dots + \bar{y}_{27} + \bar{y}_{28} - \bar{y}_{29} - \bar{y}_{30} + \bar{y}_{31} + \bar{y}_{32}) \\
 &\quad \vdots \\
 q_{ABDE} &= \frac{1}{32} (\bar{y}_1 - \bar{y}_2 - \bar{y}_3 + \bar{y}_4 + \bar{y}_5 - \bar{y}_6 - \dots - \bar{y}_{27} + \bar{y}_{28} + \bar{y}_{29} - \bar{y}_{30} - \bar{y}_{31} + \bar{y}_{32}) \\
 q_{ACDE} &= \frac{1}{32} (\bar{y}_1 - \bar{y}_2 - \bar{y}_3 + \bar{y}_4 - \bar{y}_5 + \bar{y}_6 + \dots + \bar{y}_{27} - \bar{y}_{28} + \bar{y}_{29} - \bar{y}_{30} - \bar{y}_{31} + \bar{y}_{32}) \\
 q_{BCDE} &= \frac{1}{32} (\bar{y}_1 - \bar{y}_2 - \bar{y}_3 + \bar{y}_4 - \bar{y}_5 + \bar{y}_6 + \dots + \bar{y}_{27} - \bar{y}_{28} + \bar{y}_{29} - \bar{y}_{30} - \bar{y}_{31} + \bar{y}_{32}) \\
 q_{ABCDE} &= \frac{1}{32} (-\bar{y}_1 + \bar{y}_2 + \bar{y}_3 - \bar{y}_4 + \bar{y}_5 - \bar{y}_6 - \dots + \bar{y}_{27} - \bar{y}_{28} + \bar{y}_{29} - \bar{y}_{30} - \bar{y}_{31} + \bar{y}_{32})
 \end{aligned}$$

O próximo passo consiste em calcular o percentual da variação da sobrecarga atribuído a cada um dos cinco fatores e a cada interação entre eles. Para essa finalidade, aplica-se a Soma de Quadrados Total (SS_T), a Soma dos Quadrados de todas as respostas dos experimentos (SS_Y) e as Somas de Quadrados que atribuem porções de variação a fatores-interações (SS_X) ou a erros (SS_E).

A Eq.13 mostra que SS_Y é a soma dos quadrados das respostas obtidas em todas as repetições i de todos os experimentos j .

$$SS_Y = \sum_{\substack{1 \leq i \leq 30 \\ 1 \leq j \leq 32}} y_{ij}^2 \quad (13)$$

A Eq.14 descreve a SS_T , que depende da SS_Y e do coeficiente q_0 .

$$SS_T = SS_Y - 2^k r \cdot (q_0)^2 \quad (14)$$

A Eq.15 descreve as SS_x , que podem ser atribuídas a fatores ou interações.

$$SS_x = 2^k r \cdot (q_x)^2 \quad (15)$$

A Eq.16 fornece a Soma dos Quadrados para erros experimentais.

$$SS_E = SS_Y - 2^k r \cdot (q_0^2 + q_A^2 + q_B^2 + \dots + q_{ABCDE}^2) \quad (16)$$

Por fim, a Eq.17 revela o percentual da variação da sobrecarga atribuído a fatores, interações ou erros.

$$\text{Porção de Variação} = \frac{SS_x}{SS_T} \text{ ou } \frac{SS_E}{SS_T} \quad (17)$$

4.3 Avaliação dos Fatores

Os fatores da Tabela 1 foram analisados em 32 experimentos de um modelo Fatorial $2^k r$. Foram realizadas $r = 30$ repetições para cada experimento. Este número de repetições é suficiente para informar quais fatores são os mais significativos na variação da resposta, que é a sobrecarga na rede. O algoritmo testado foi o QoN++. Como os instantes de conexão entre nós foram registrados em uma base de dados, a criação de diferentes mensagens em instantes de tempo variáveis é controlada por sementes de um número aleatório. Resultados na Tabela 4.

Tabela 4: Primeira avaliação de fatores

Fator	A	B	C	D	E	AB	AC	AD
Porções de Variação %	0.56	26.76	0.04	17.33	33.03	0.06	0.01	0.15
Fator	AE	BC	BD	BE	CD	CE	DE	ABC
Porções de Variação %	3.87	0.01	0.55	6.31	0.10	0.04	0.16	0.10
Fator	ABD	ABE	ACD	ACE	ADE	BCD	BCE	BDE
Porções de Variação %	3.44	0.71	0.02	0.01	0.40	0.02	0.03	0.30
Fator	CDE	ABCD	ABCE	ABDE	ACDE	BCDE	ABCDE	Erros
Porções de Variação %	0.00	0.03	0.03	2.25	0.03	0.02	0.03	3.64

Os fatores B (tamanho do *buffer*), D (intervalo de geração de mensagens) e E (número de cópias) são os que mais influenciam a variação nos resultados da sobrecarga. Todos esses fatores possuem porções maiores ou iguais a 5% e são relevantes para os estudos da redução da sobrecarga em rede. A interação entre os fatores B e E também apresentou um elevado percentual. Para investigar melhor esses resultados, foi feito um segundo conjunto de experimentos levando-se em consideração apenas os fatores B, D e E, bem como suas interações. Nesta segunda avaliação, os valores dos fatores A e C foram mantidos fixos em A = 400 min e C = 40h. Os valores da avaliação estão na Tabela 5.

Tabela 5: Segunda avaliação de fatores

Fator	B	D	E	BD	BE	DE	BDE	Erros
Porções de Variação %	32.99	15.49	41.93	0.00	3.32	0.18	2.76	3.32

Os fatores B (tamanho do *buffer*), D (intervalo de geração de mensagens) e E (número de cópias) permanecem sendo os mais influentes na variação da sobrecarga. Ressalta-se que o percentual atribuído aos erros permanece inferior a 5%. Como não é possível simplificar ainda mais o teste, as avaliações de fatores foram concluídas.

5. AVALIAÇÃO DE DESEMPENHO

Neste capítulo, será avaliado o desempenho do QoN++ na redução da sobrecarga na rede, na melhoria das taxas de entrega e na manutenção de um atraso médio competitivo em relação aos algoritmos Epidêmico, P_{Ro}PHET, *Spray and Wait* e QoN-ASW. Em vez de cinco fatores, apenas os três fatores aprovados no capítulo anterior serão modificados em um maior número de níveis, em experimentos conhecidos como “um fator por vez”.

Para os cálculos da sobrecarga, da taxa de entrega e do atraso médio, foram consideradas as participações de todos os nós do cenário. A taxa de entrega é a razão entre o número de mensagens entregues (N_{ent}) e o número de mensagens criadas (N_{cr}) durante a simulação, fornecida pela Eq. 18. O valor está situado em uma escala de 0 a 1.

$$Taxa\ de\ Entrega = \frac{N_{ent}}{N_{cr}} \quad (18)$$

O atraso médio (em segundos) corresponde à razão entre o somatório dos tempos de entrega (ou atrasos At_x) de todas as mensagens e o número de mensagens entregues (N_{ent}) aos destinos. É dado pela Eq. 19. Representa o tempo médio necessário para o transporte de uma mensagem do nó gerador ao nó de destino. Quanto mais baixo for o atraso, mais rápida é a troca de mensagens.

$$Atraso\ Médio = \frac{\sum_{x=1}^{N_{ent}} At_x}{N_{ent}} \quad (19)$$

Os desempenhos dos algoritmos de roteamento para o cenário do Infocom06 foram obtidos através de simulações computacionais, que configuram os valores dos fatores em cada experimento.

Para cada teste de desempenho, os níveis de um fator foram alterados, enquanto os demais fatores mantiveram-se fixos. As sobrecargas de rede causadas pelos algoritmos são ilustradas em gráficos de barras. Os resultados das taxas de entrega e do atraso são representados por gráficos de linhas, onde os pontos correspondem

a uma média aritmética das respostas fornecidas por um algoritmo para um determinado conjunto de configurações de teste.

Os pontos dos gráficos de linhas estão acompanhados por intervalos de confiança de 95%, calculados de acordo com a Eq.20.

$$\text{Intervalo de confiança} = \bar{y} \pm 1,96 \left(\frac{\sigma}{\sqrt{r}} \right) \quad (20)$$

onde \bar{y} é a média das respostas y_i após r repetições e σ corresponde ao desvio padrão das amostras [25], dado pela Eq. 21.

$$\sigma = \sqrt{\sum_{i=1}^r \frac{(y_i - \bar{y})^2}{r - 1}} \quad (21)$$

A constante 1,96 indica que 95% da área sob a curva de distribuição normal das amostras está dentro de 1,96 desvios-padrões da média.

5.1 Tamanho de *Buffer* variável

O primeiro teste de desempenho dos algoritmos contém oito diferentes níveis para o tamanho de *buffer*. Os outros fatores relevantes para a variação, como intervalo de geração de mensagens e número de cópias, terão valores fixos.

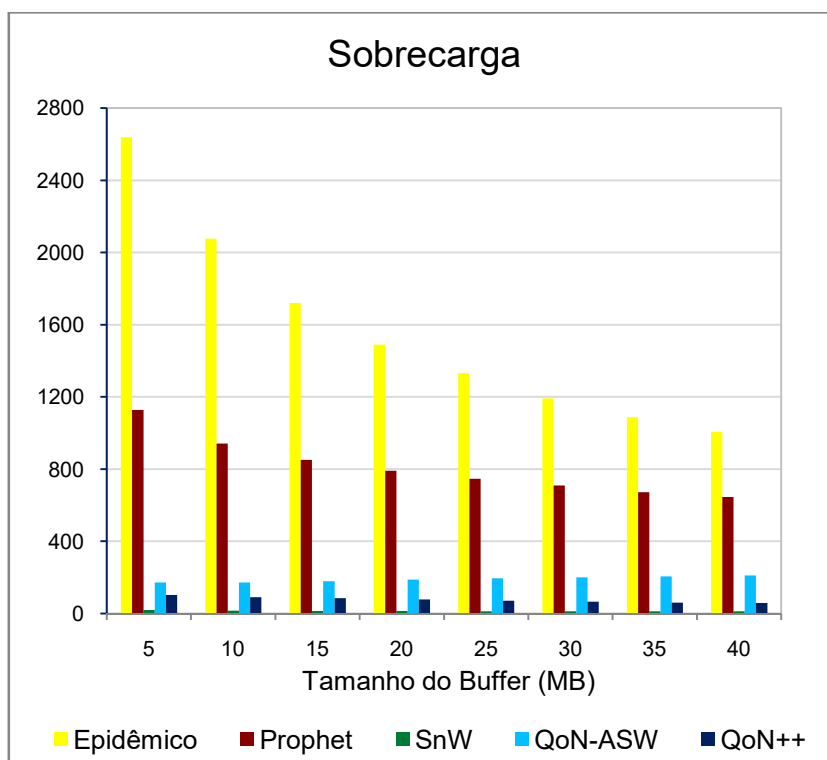
Ressalta-se que as interações entre os diferentes fatores não precisam ser avaliadas à parte, porque as porções de variação de sobrecarga na rede a elas atribuídas foram inferiores a 5% e são muito menores que as porções de cada fator isolado, conforme a Tabela 5.

Foram adotadas as configurações da Tabela 6 como base para a simulação. A única variável alterada foi o tamanho do *buffer* dos nós, que assumiu valores no conjunto $Buffer = \{5, 10, 15, 20, 25, 30, 35, 40\}$, em MBytes.

Tabela 6: Configurações da simulação 5.1 - *Buffer* variável

Tempo total de simulação (h)	40
Intervalo de geração de mensagens (s)	60
Time to Live (TTL) da mensagem (min)	400
Velocidade de transmissão (kBps)	250
Range de transmissão (m)	10
Tamanho das mensagens (kB)	[300,500]
Número de cópias - SnW/QoN-ASW/QoN++	10
Número de slots - QoN-ASW/QoN++	40
Compartilhamento - SnW	Binário

Cada algoritmo foi executado seguindo as configurações da Tabela 6 e os valores do conjunto *Buffer*. Os resultados da sobrecarga em rede são ilustrados nos Gráficos 1-2. As taxas de entrega de mensagens são fornecidas pelo Gráfico 3. Os valores de atraso médio são dados no Gráfico 4.

Gráfico 1 – Sobrecarga em rede para tamanho de *Buffer* variável

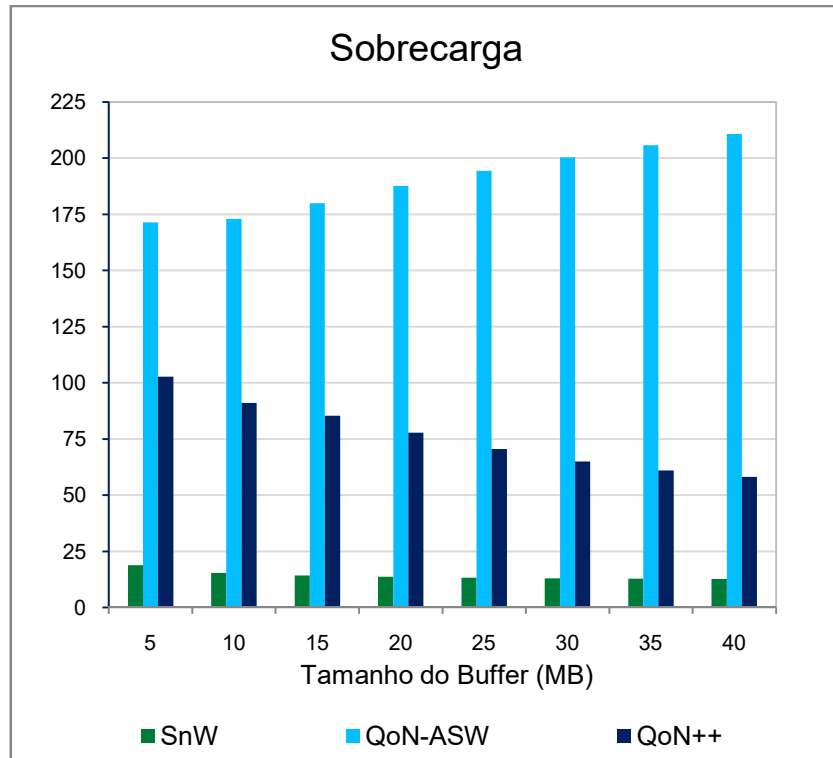


Gráfico 2 - Sobrecarga em rede para tamanho de *Buffer* variável (SnW, QoN-ASW e QoN++)

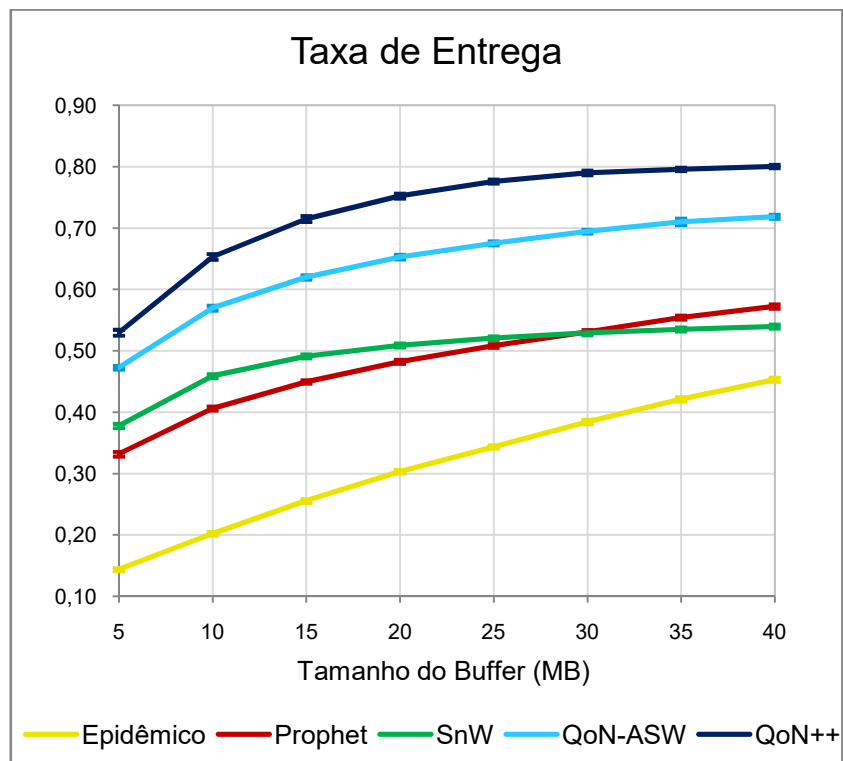


Gráfico 3 - Taxas de entrega para tamanho de *Buffer* variável

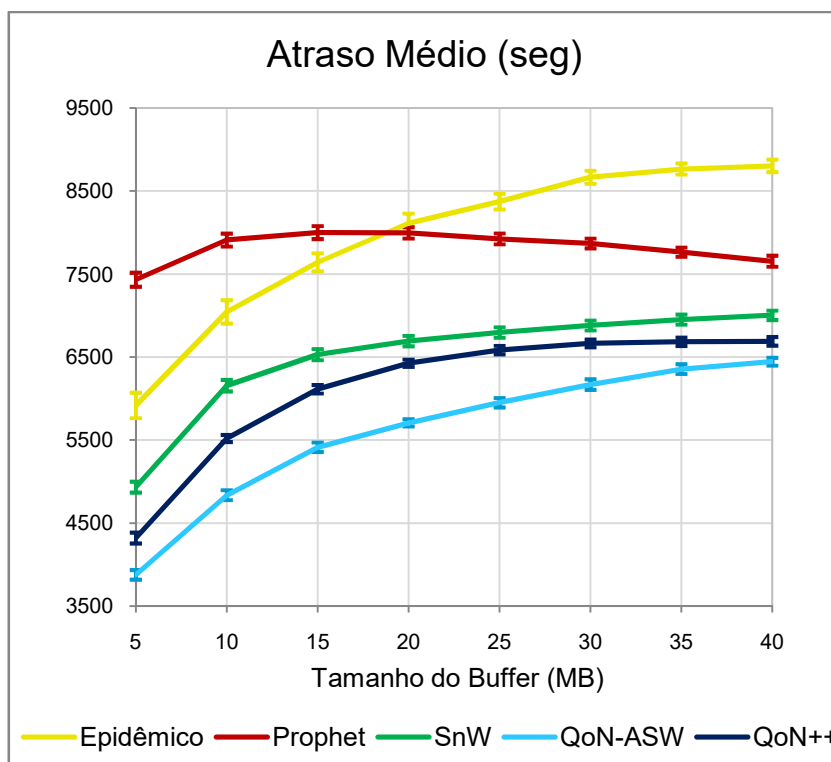


Gráfico 4 - Atraso médio para tamanho de *Buffer* variável

O aumento do *buffer* permite aos nós um maior armazenamento de mensagens. Isso significa que os algoritmos - principalmente os do tipo *flooding*, como o Epidêmico - aumentam suas capacidades de difusão de dados na rede, elevando suas taxas de entrega. À medida que o número de entregas cresce, a sobrecarga tende a ser reduzida. No entanto, as mensagens que só puderam ser entregues após o aumento do *buffer* tiveram as suas latências computadas no cálculo do atraso médio, fazendo com que o Epidêmico apresente os maiores atrasos dentre todos os algoritmos.

As funções de probabilidade utilizadas pelo P_{Ro}PHET orientam o encaminhamento de mensagens a nós com maiores chances de encontrar o destino. Elas reduzem a sobrecarga a valores sempre inferiores aos do algoritmo Epidêmico e tornam os encaminhamentos mais eficientes, aumentando a taxa de entrega e acarretando as menores variações no atraso médio.

O *Spray And Wait* possui uma fase *Wait* passiva, pois os nós são instruídos a encaminhar a mensagem diretamente ao destino. Caso não consigam, eles devem aguardar até que haja um contato direto com o receptor final, antes que haja um *overflow* em suas memórias ou que o TTL da mensagem expire. Como a quantidade de repasses é limitada pelo número máximo de cópias da mensagem, o SnW

apresenta os menores valores de sobrecarga na rede, dentre todos os algoritmos. Porém, a dependência de contatos diretos faz com que as taxas de entrega se estabilizem após um dado incremento no tamanho do *buffer*. Conseqüentemente, o atraso médio também tende a apresentar poucas variações, especialmente para *buffers* de 15MB ou mais.

O QoN-ASW mescla elementos do P_{Ro}PHET e do SnW. Sua fase *Wait* é ativa, onde um nó é instruído a repassar sua cópia da mensagem caso o outro nó do contato apresente maiores probabilidades de encontrar o destino ou algum intermediário em potencial. Sua sobrecarga é maior que a do SnW e tende a crescer com uma maior disponibilidade de memória. No entanto, as funções de probabilidade modificadas do P_{Ro}PHET operam em conjunto com o controle de cópias do SnW, reduzindo os riscos de saturação dos *buffers* dos nós, elevando as taxas de entrega e apresentando os menores atrasos.

Para o QoN++, as entregas de mensagens são eventos que devem ser notificados aos demais nós da rede. Uma vez que estes nós sejam informados, eles descartam as mensagens dos seus *buffers* e informam outros nós sobre os eventos de entrega. Como consequência, a sobrecarga tende a diminuir com o aumento do *buffer*, um efeito contrário ao que é visto no QoN-ASW. Para 40MB, a sobrecarga do QoN++ corresponde a apenas 27% da sobrecarga do QoN-ASW. A justificativa é que o aumento do *buffer* favorece um incremento na taxa de entrega, conforme visto nos demais algoritmos. Desse modo, o número de notificações na rede cresce com o aumento do *buffer* e os nós serão informados sobre tais eventos com maior antecedência, reduzindo o número de repasses na rede.

Outra característica do QoN++ é a mudança de estratégia na distribuição de cópias. Apesar de a qualidade do nó ser um valor essencial para a escolha da estratégia, as cópias apenas serão repartidas com base no valor numérico da qualidade se esta estiver em uma faixa média dentro do intervalo $[0,1]$, mais especificamente em $[0.3, 0.6)$. Se a qualidade do nó apresentar um valor muito próximo de zero ou um, o número de cópias que poderão ser mantidas ou encaminhadas é mínimo.

A interpretação é que um nó com uma qualidade muito baixa só poderá ser útil como um intermediário de outro nó que, ocasionalmente, seja um bom candidato para entregar a mensagem ao destino. Para 20MB, o aumento na taxa de entrega é de 15%. Essa vantagem é compensada por um aumento no atraso médio. Vale

ressaltar que o atraso do QoN++ continua sendo menor que os valores do Epidêmico, PProPHET e SnW.

5.2 Intervalo de geração de mensagens variável

Para o segundo teste de desempenho, oito níveis serão atribuídos ao intervalo de geração. Agora, o tamanho de *buffer* e o número de cópias terão valores fixos. No último teste, o valor de 40MB para *buffer* apresentou a mais baixa sobrecarga, a mais alta taxa de entrega e um atraso um pouco maior que o do QoN-ASW. Portanto, esse valor será fixado para o tamanho de *buffer*.

Foram adotadas as configurações da Tabela 7 como base para a simulação. A única variável alterada foi o intervalo de geração das mensagens, que assumiu valores no conjunto Intervalo = {30, 40, 50, 60, 70, 80, 90, 100}, em segundos.

Tabela 7: Configurações da simulação 5.2 – Intervalo de geração variável

Tempo total de simulação (h)	40
Tamanho do <i>Buffer</i> (MB)	40
Time to Live (TTL) da mensagem (min)	400
Velocidade de transmissão (kBps)	250
Range de transmissão (m)	10
Tamanho das mensagens (kB)	[300,500]
Número de cópias - SnW/QoN-ASW/QoN++	10
Número de slots - QoN-ASW/QoN++	40
Compartilhamento - SnW	Binário

Cada algoritmo foi executado seguindo as configurações da Tabela 7 e os valores do conjunto Intervalo. Os resultados da sobrecarga em rede são ilustrados nos Gráficos 5-6. As taxas de entrega de mensagens são fornecidas pelo Gráfico 7. Os valores de atraso médio são dados no Gráfico 8.

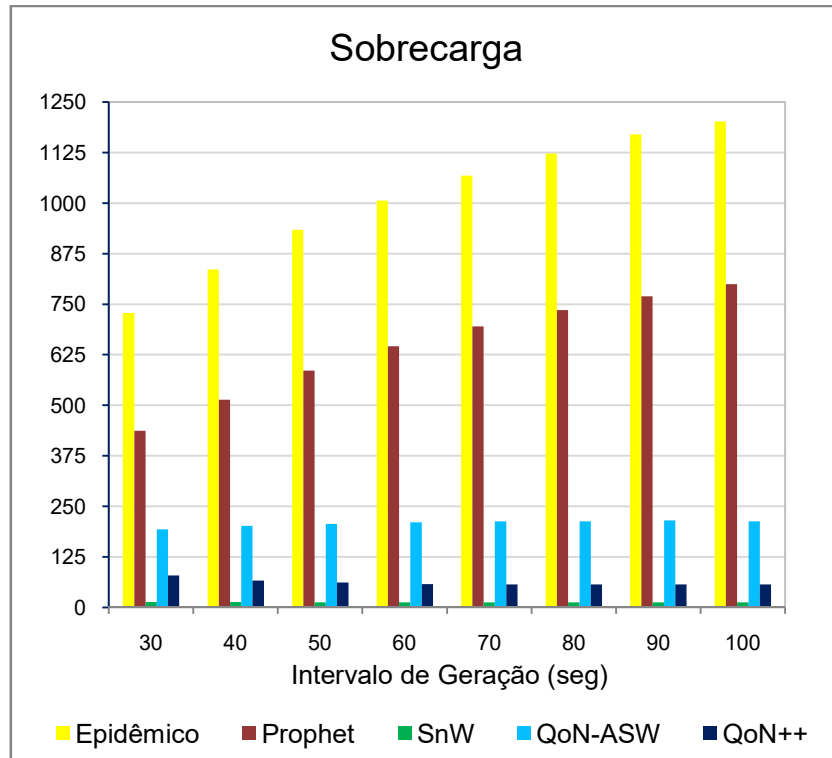


Gráfico 5 – Sobrecarga em rede para intervalo de geração de mensagens variável

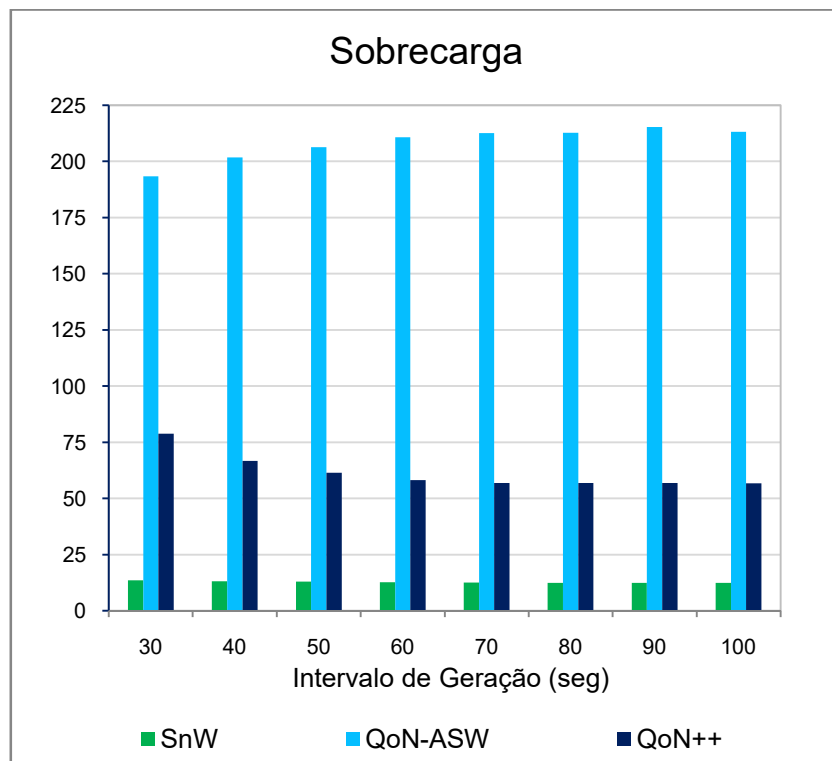


Gráfico 6 - Sobrecarga em rede para intervalo de geração variável (SnW, QoN-ASW e QoN++)

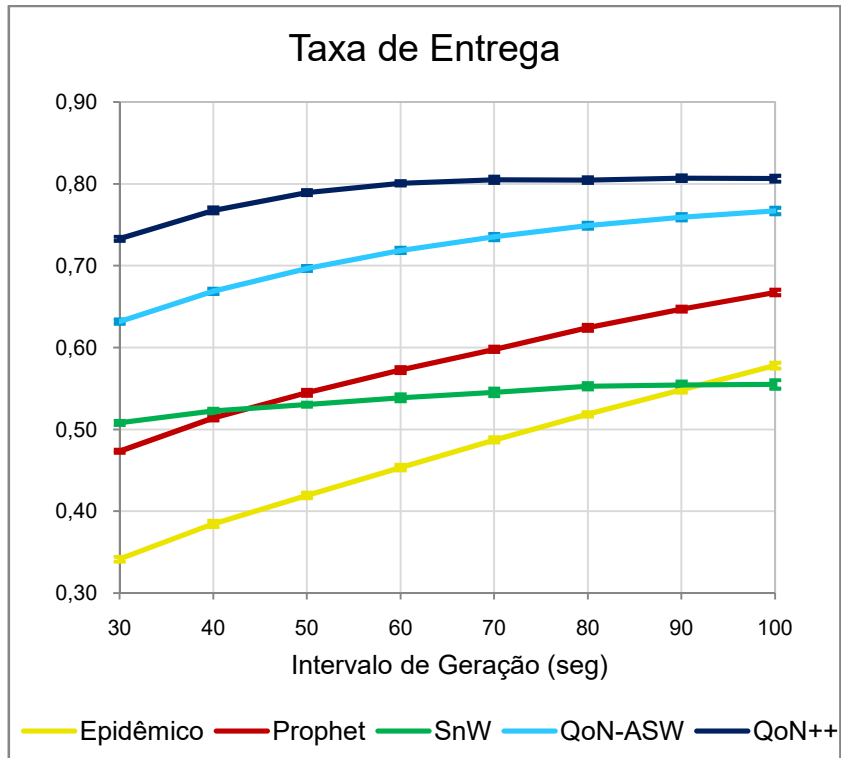


Gráfico 7 - Taxas de entrega para intervalo de geração variável

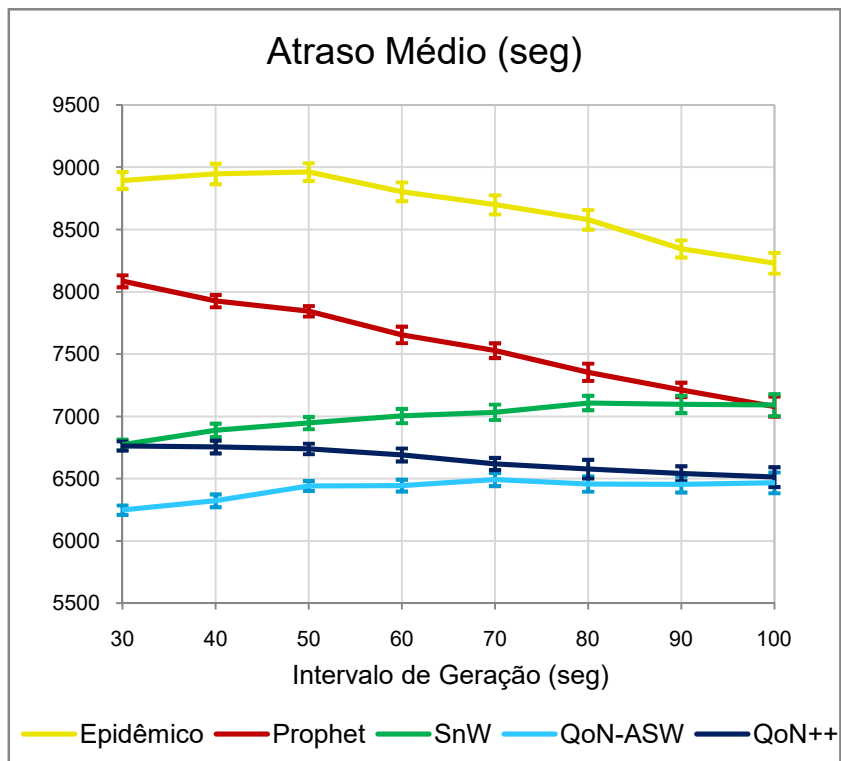


Gráfico 8 - Atraso médio para intervalo de geração variável

O aumento do intervalo de geração de mensagens no cenário causa a redução do número de mensagens em circulação na rede. Desse modo, os recursos de memória dos nós possuem menor probabilidade de entrar em saturação, prevenindo que mensagens sejam descartadas por nós com *buffers* lotados. Isso favorece diretamente os algoritmos que não possuem controle do número máximo de cópias de mensagens na rede, como Epidêmico e P_{Ro}PHET. As taxas de entrega destes algoritmos apresentam as maiores inclinações de crescimento e os valores de atrasos tendem ser reduzidos, pois ocorre uma proliferação de cópias na rede acompanhada de um menor risco de descarte. No entanto, com menores riscos de saturação da rede, a quantidade de repasses apresenta grande crescimento, elevando as sobrecargas do Epidêmico e do P_{Ro}PHET.

O SnW depende de contatos diretos para realizar entregas durante a fase *Wait*. Se o número de mensagens na rede for reduzido, a interferência nas taxas de entrega e nos atrasos será modesta. Como esperado, ele possui as menores sobrecargas, mas suas taxas de entregas tendem a ser menos eficientes se comparadas com as de outros algoritmos, ficando estagnadas.

O QoN-ASW e o QoN++ possuem controle do número máximo de cópias de uma mensagem na rede. Essa característica ajuda a manter as sobrecargas de ambos sem grandes alterações. Suas taxas de entrega e seus atrasos tendem a se aproximar à medida que o intervalo de geração se torna mais prolongado.

Destaca-se a vantagem da sobrecarga do QoN++ sobre a resultante do QoN-ASW, correspondendo a 27% do valor desta para um intervalo de geração igual a 60 segundos ou mais.

5.3 Número de cópias variável

Para o terceiro teste de desempenho, oito níveis foram atribuídos ao número de cópias criadas pelo nó gerador de uma mensagem. O tamanho de *buffer* teve o valor de 40MB mantido. Para o intervalo de geração, foi adotado novamente o valor de 60 segundos, à semelhança do experimento 5.1, com *buffers* variáveis.

Foram adotadas as configurações da Tabela 8 como base para a simulação. A única variável alterada foi o número de cópias das mensagens, que assumiu valores no conjunto Cópias = {2, 6, 10, 14, 18, 22, 26, 30}.

Tabela 8: Configurações da simulação 5.3 – número de cópias variável

Tempo total de simulação (h)	40
Tamanho do <i>Buffer</i> (MB)	40
Time to Live (TTL) da mensagem (min)	400
Velocidade de transmissão (kBps)	250
Range de transmissão (m)	10
Tamanho das mensagens (kB)	[300,500]
Intervalo de geração de mensagens (seg)	60
Número de slots - QoN-ASW/QoN++	40
Compartilhamento - SnW	Binário

Cada algoritmo foi executado seguindo as configurações da Tabela 8 e os valores do conjunto Cópias. Os resultados da sobrecarga em rede são ilustrados nos Gráficos 9-10. As taxas de entrega de mensagens são fornecidas pelo Gráfico 11. Os valores de atraso médio estão presentes no Gráfico 12.

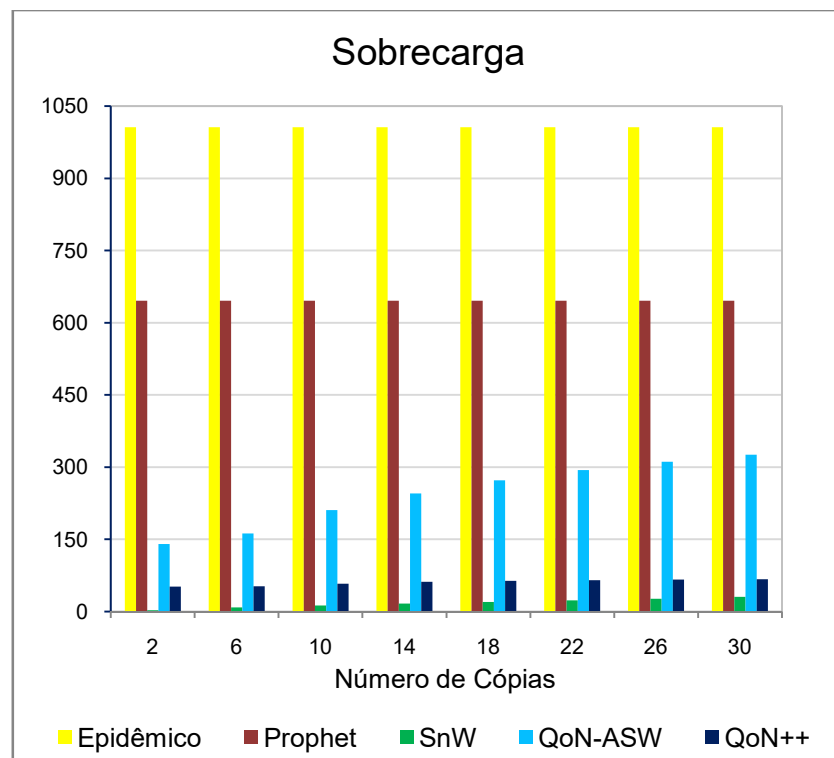


Gráfico 9 – Sobrecarga em rede para número de cópias variável

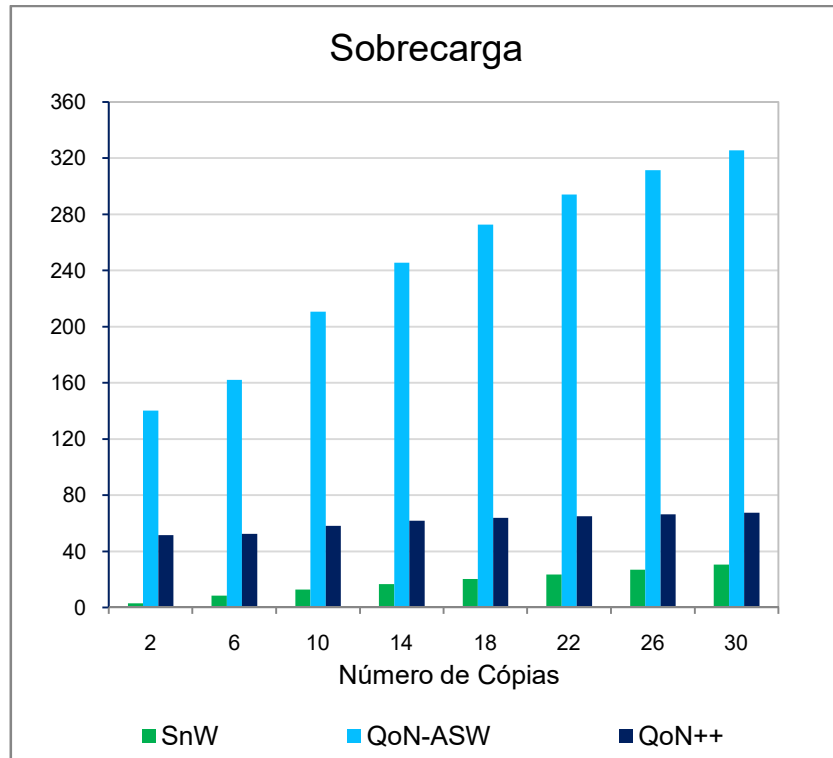


Gráfico 10 - Sobrecarga em rede para número de cópias variável (SnW, QoN-ASW e QoN++)

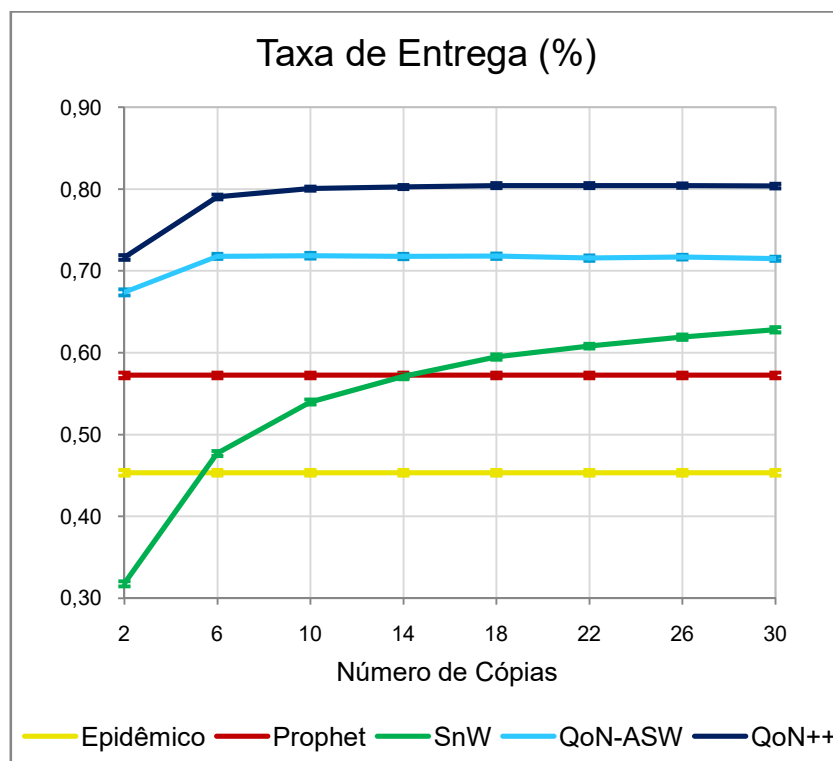


Gráfico 11 - Taxas de entrega para número de cópias variável

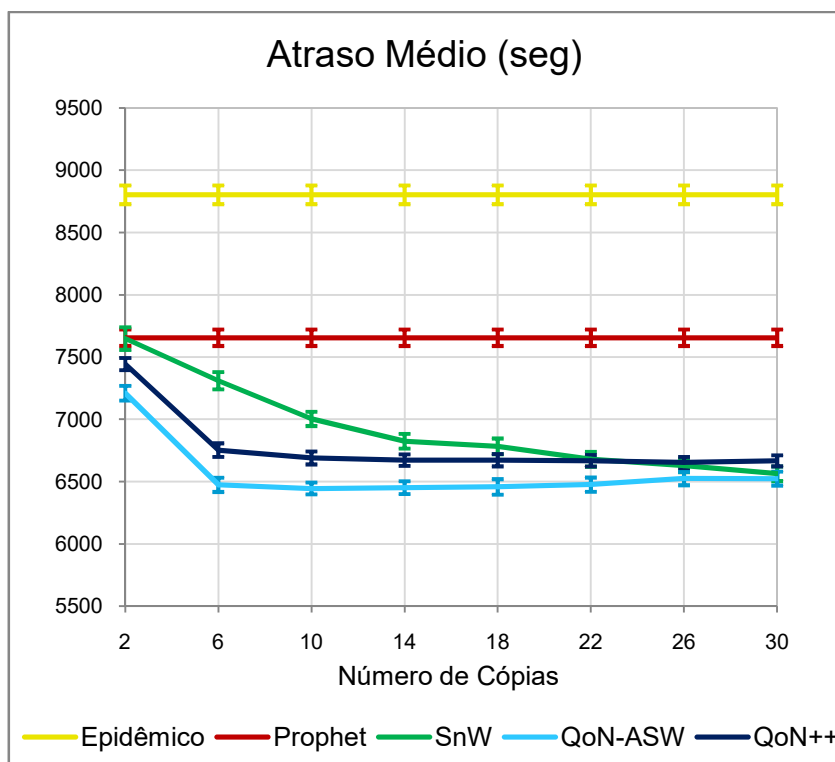


Gráfico 12 - Atraso médio para número de cópias variável

Os algoritmos Epidêmico e P_{Ro}PHET não possuem um controle do número máximo de cópias de uma mensagem na rede. Portanto, seus valores para sobrecarga, taxa de entrega e atraso não estão sujeitos a mudanças. Eles são definidos por outras configurações do experimento, informadas na Tabela 8.

O *Spray And Wait* é dependente do número de cópias em sua essência. O aumento deste número é uma solução para elevar a eficiência do SnW. Sua taxa de entrega foi a que apresentou a maior evolução dentre todas, embora sejam inferiores às taxas de QoN-ASW e QoN++. Os atrasos foram reduzidos porque, como a entrega é direta na fase *Wait*, as chances de as mensagens chegarem mais rápido a seus destinos é diretamente proporcional à quantidade de cópias espalhadas pela rede. Quanto maior for o número de cópias, maior será a taxa de entrega e menor será o atraso. No entanto, o número de cópias é um fator capaz de elevar consideravelmente a sobrecarga do SnW, pois o número de encaminhamentos na fase *Spray* sempre cresce ou diminui em função do aumento ou redução do número de cópias iniciais.

O QoN-ASW e o QoN++ se beneficiam das funções de probabilidade herdadas do P_{Ro}PHET e do número controlado de cópias do SnW para encaminhar as mensagens aos nós com maior potencial de entrega. O QoN++ escolhe a estratégia

de repasse em função do valor da qualidade do nó e emprega tal valor no cálculo matemático das cópias para repasse, caso esteja inserido em uma faixa de valores médios entre zero e um. À semelhança dos experimentos anteriores, evidencia-se que essa abordagem oferece ao QoN++ as maiores taxas de entrega e um atraso que, a partir de 22 cópias de mensagem, está em uma faixa de valores próxima às latências de QoN-ASW e SnW.

No entanto, o comportamento mais interessante é visualizado no gráfico de sobrecargas. Por falta de um mecanismo de notificação de entregas, a sobrecarga do QoN-ASW cresce até chegar à metade do valor do P_{RoPHET} e a 1/3 do valor do Epidêmico, para o número de cópias igual a 30. Para esse mesmo número de cópias, a sobrecarga na nossa proposta, QoN++, equivale a apenas 20% da sobrecarga do QoN-ASW.

6. CONCLUSÕES E SUGESTÕES

O QoN++ é uma modificação do novo algoritmo QoN-ASW que trata a entrega de mensagens a nós de uma rede como eventos que devem ser notificados aos demais nós. Após receber a informação a respeito de uma determinada entrega, os nós atualizam seus *buffers* e repassam esse conhecimento aos demais contatos. Como resultado, tem-se a constante redução de sobrecarga na rede em relação ao desempenho mostrado por QoN-ASW.

Em paralelo, o QoN++ também encaminha suas mensagens utilizando o conceito de Qualidade dos Nós para elevar as taxas de entrega e para manter os atrasos em níveis competitivos em relação aos demais algoritmos analisados. Para essa finalidade, ele ajusta sua estratégia em função das qualidades dos nós que queiram compartilhar mensagens. Se um dos nós tiver a qualidade muito baixa, ele receberá ou manterá uma quantidade mínima fixa de cópias, atuando como um intermediário entre nós com maior potencial para encontrar os nós de destino.

Durante as análises experimentais, o QoN++ apresentou as melhores taxas de entrega em uma comparação que incluía os amplamente conhecidos algoritmos Epidêmico, PRoPHET e *Spray And Wait*, além do moderno QoN-ASW. No ranking dos atrasos médios, o QoN++ só perde para a agilidade do QoN-ASW, vencendo os demais algoritmos em quase todas as simulações. Para compensar, a sobrecarga em QoN++ pode ser até quatro vezes menor que a de QoN-ASW para diversas configurações de teste.

O QoN++ é de particular utilidade em cenários onde se deseja uma elevada taxa de entrega de pacotes com uma baixa quantidade de sobrecarga na DTN, ajudando os dispositivos móveis a economizar memória, bateria e uso de dados de rede.

6.1 Trabalhos Futuros

A qualidade dos nós é um conceito que depende, em parte, de funções de cálculo de probabilidade de encontro entre nós. O comportamento social dos nós pode ser utilizado na detecção de comunidades. Como sugestão de pesquisa futura, recomenda-se a implementação de algoritmos de clusterização, como o *k-means* [26], Esperança-Maximização (EM) [27] ou modulação de Louvain [28]. O objetivo é

reduzir o número de nós que precisam realizar encaminhamentos de mensagens, através da escolha de nós específicos destas comunidades. Os nós selecionados deverão executar o QoN++. Desse modo, pode-se verificar como o uso em conjunto do QoN++ com técnicas de clusterização pode afetar a sobrecarga e a taxa de entrega em redes.

6.2 Publicação

Obteve-se uma publicação após o desenvolvimento do QoN++.

- Rosa, W. C.; Mota, E. S.; Rezende, J. F.; Carvalho, C. B. - "QoN++: Roteamento por Qualidade dos Nós com redução de Overhead em Redes DTN.", XXXVIII Simpósio Brasileiro de Telecomunicações e Processamento de Sinais - SBrT'2020, Florianópolis, SC, Brasil, Novembro 2020.

7. REFERÊNCIAS BIBLIOGRÁFICAS

[1] CARVALHO, C. B.; MOTA, E. S.; FERRAZ, E.; SEIXAS, P.; SOUZA, P.; TAVARES, V.; FILHO, W. L.; FERREIRA, D.; MANZONI, P.; CALAFATE, C. *Entropy based routing for mobile, low Power and lossy wireless sensors networks*. *International Journal of Distributed Sensor Networks*, Vol.15 (7), 2019.

[2] NEVES, E. V.; MARTINS, R.; CARVALHO, C. B.; MOTA, E. S. *Disseminação de mensagens DTN com base em grupos de interesses*. In: ERIN, 2017, MANAUS. ERIN 2017 - IV ESCOLA REGIONAL DE INFORMÁTICA, 2017. v. 2017.

[3] SOARES, D.; MATTHAUS, B.; MOTA, E. S.; CARVALHO, C. B. *A Feasibility Study of Watchdogs on Opportunistic Mobile Networks*. In: IEEE ISCC 2018, 2018, Natal, Brazil. IEEE Symposium on Computers and Communications, 2018. v. 23.

[4] SHAHID, A.; KHALID, B.; SHAUKAT, S.; ALI, H.; QADRI, M. Y. *Internet of Things Shaping Smart Cities: A Survey*. *Internet of Things and Big Data Analytics Toward Next-Generation Intelligence*, Springer International Publishing AG, 2018.

[5] JOYIA, G. J.; LIAQAT, R. M.; FAROOQ, A.; REHMAN, S. *Internet of Medical Things (IOMT): Applications, Benefits and Future Challenges in Healthcare Domain*, *Journal of Communications*, Vol. 12, No. 4, 2017

[6] ROMAN, D. H.; CONLEE, K. D. *The Digital Revolution comes to US Healthcare*. *Internet of Things*, Vol. 5, Ed. Goldman Sachs, 2015.

[7] BOYES, H.; HALLAQ, B.; CUNNINGHAM, J.; WATSON, T. *The Industrial Internet of Things (IIoT): An Analysis Framework*. *Computers in Industry*, Vol. 101, Ed. Elsevier, 2018.

[8] DAUGHERTY, P.; NEGM, W.; BANERJEE, P.; ALTER, A. *Driving Unconventional Growth through the Industrial Internet of Things*. Accenture Technology, 2016.

[9] ZHANG, Q. *Precision Agriculture Technology for Crop Farming*. CRC Press, 2015.

- [10] XIAO, L.; WANG, Z. *Internet of Things: a New Application for Intelligent Traffic Monitoring System*. Journal of Networks, Vol. 6, No. 6, 2011.
- [11] PERKINS, C. E.; ROYER, E. M. *Ad-hoc On-Demand Distance Vector Routing*. The Second IEEE Workshop on Mobile Computing Systems and Applications, 1999.
- [12] JOHNSON, D. B.; MALTZ, D. A. *Dynamic Source Routing in Ad Hoc Wireless Networks*. Mobile Computing, Kluwer Academy, 1996.
- [13] VAHDAT, A.; BECKER, D. *Epidemic routing for partially connected ad hoc networks*. Department of Computer Science, Duke University, 2000.
- [14] LINDGREN, A.; DORIA, A.; SCHELÉN, O. *Probabilistic routing in intermittently connected networks*. Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc, 2003.
- [15] SPYROPOULOS, T.; PSOUNIS, K.; RAGHAVENDRA, C. S. *Spray and Wait: An efficient routing scheme for intermittently connected mobile networks*. Proceeding of the 2005 ACM SIGCOMM workshop on Delay-tolerant networking, 2005.
- [16] CUI, J.; CAO, S.; CHANG, Y.; WU, L.; LIU, D.; YANG, Y. *An Adaptive Spray and Wait Routing Algorithm Based on Quality of Node in Delay Tolerant Network*. IEEE Access, April 2019.
- [17] CHAINTREAU, A.; HUI, P.; CROWCROFT, J.; DIOT, C.; GASS, R.; SCOTT, J. *Impact of human mobility on opportunistic forwarding algorithms*. IEEE Trans. Mobile Comput., Vol. 6, No. 6, 2007.
- [18] KERÄNEN, A.; OTT, J.; KÄRKKÄINEN, T. *The ONE Simulator for DTN protocol evaluation*. In Proc. Int. Conf. Simulation TOOLS Techn., Rome, Italy, 2009.
- [19] MISRA, S.; SAHA, B. K.; PAL, S. *Opportunistic Mobile Networks - Advances and Applications*. Ed. Springer, 2016.
- [20] WANG, G.; WANG, B.; GAO, Y. *Dynamic Spray and Wait Routing algorithm with Quality of Nodes in Delay Tolerant Network*. 2010 International Conference on Communications and Mobile Computing.

[21] DU, X.; LAI, W.; ZHANG, J.; WANG, H. *A motion and counting based-routing algorithm for delay tolerant network*. International Journal of Communication, Network and System Sciences, Vol. 10, 2017.

[22] GUAN, J.; CHU, Q.; YOU, I. *The social relationship based adaptive multi-spray-and-wait routing algorithm for disruption tolerant network*. Mobile Information Systems, n° 1819495, 2017.

[23] WANG, G.; ZHANG, H.; PAN, D.; TAO, J. *A routing scheme based on spray-and-wait and connecting-capability of node*. International Conference on Progress in Informatics and Computing, Nanjing, China, 2016.

[24] MONTGOMERY, D. C.; RUNGER, G. C. *Applied Statistics and Probability for Engineers, 7th Ed*. Wiley Books, 2018.

[25] WALPOLE, R. E.; MYERS, R. H.; MYERS, S. L.; YE, K. *Probability & Statistics for Engineers & Scientists, 9th Ed*. Prentice Hall, 2012.

[26] HARTIGAN, J. A.; WONG, M. A, "Algorithm AS 136: *A k-means clustering algorithm*" Journal of the Royal Statistical Society. Series C (Applied Statistics), vol. 28, no. 1, pp. 100 – 108, 1979. 16, 30, 41

[27] McLACHLAN G.; KRISHNAN T. *The EM algorithm and extensions*. John Wiley & Sons, 2007, vol. 382. 29

[28] BLONDEL, V. D.; GUILLAUME, J-L.; LAMBIOTTE, R.; LEFEBVRE, E. "*Fast unfolding of communities in large networks*". Journal of Statistical Mechanics: Theory and Experiment. 2008 (10): P10008.

APÊNDICE A

As equações completas para cada experimento da avaliação dos fatores são fornecidas a seguir:

$$\begin{aligned} \bar{y}_1 = & q_0 - q_A - q_B - q_C - q_D - q_E + q_{AB} + q_{AC} + q_{AD} + q_{AE} + q_{BC} + q_{BD} + q_{BE} + q_{CD} + \\ & q_{CE} + q_{DE} - q_{ABC} - q_{ABD} - q_{ABE} - q_{ACD} - q_{ACE} - q_{ADE} - q_{BCD} - q_{BCE} - q_{BDE} - q_{CDE} + \\ & q_{ABCD} + q_{ABCE} + q_{ABDE} + q_{ACDE} + q_{BCDE} - q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \bar{y}_2 = & q_0 - q_A - q_B - q_C - q_D + q_E + q_{AB} + q_{AC} + q_{AD} - q_{AE} + q_{BC} + q_{BD} - q_{BE} + q_{CD} - \\ & q_{CE} - q_{DE} - q_{ABC} - q_{ABD} + q_{ABE} - q_{ACD} + q_{ACE} + q_{ADE} - q_{BCD} + q_{BCE} + q_{BDE} + q_{CDE} + \\ & q_{ABCD} - q_{ABCE} - q_{ABDE} - q_{ACDE} - q_{BCDE} + q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \bar{y}_3 = & q_0 - q_A - q_B - q_C + q_D - q_E + q_{AB} + q_{AC} - q_{AD} + q_{AE} + q_{BC} - q_{BD} + q_{BE} - q_{CD} + \\ & q_{CE} - q_{DE} - q_{ABC} + q_{ABD} - q_{ABE} + q_{ACD} - q_{ACE} + q_{ADE} + q_{BCD} - q_{BCE} + q_{BDE} + q_{CDE} - \\ & q_{ABCD} + q_{ABCE} - q_{ABDE} - q_{ACDE} - q_{BCDE} + q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \bar{y}_4 = & q_0 - q_A - q_B - q_C + q_D + q_E + q_{AB} + q_{AC} - q_{AD} - q_{AE} + q_{BC} - q_{BD} - q_{BE} - q_{CD} - \\ & q_{CE} + q_{DE} - q_{ABC} + q_{ABD} + q_{ABE} + q_{ACD} + q_{ACE} - q_{ADE} + q_{BCD} + q_{BCE} - q_{BDE} - q_{CDE} - \\ & q_{ABCD} - q_{ABCE} + q_{ABDE} + q_{ACDE} + q_{BCDE} - q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \bar{y}_5 = & q_0 - q_A - q_B + q_C - q_D - q_E + q_{AB} - q_{AC} + q_{AD} + q_{AE} - q_{BC} + q_{BD} + q_{BE} - q_{CD} - \\ & q_{CE} + q_{DE} + q_{ABC} - q_{ABD} - q_{ABE} + q_{ACD} + q_{ACE} - q_{ADE} + q_{BCD} + q_{BCE} - q_{BDE} + q_{CDE} - \\ & q_{ABCD} - q_{ABCE} + q_{ABDE} - q_{ACDE} - q_{BCDE} + q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \bar{y}_6 = & q_0 - q_A - q_B + q_C - q_D + q_E + q_{AB} - q_{AC} + q_{AD} - q_{AE} - q_{BC} + q_{BD} - q_{BE} - q_{CD} + \\ & q_{CE} - q_{DE} + q_{ABC} - q_{ABD} + q_{ABE} + q_{ACD} - q_{ACE} + q_{ADE} + q_{BCD} - q_{BCE} + q_{BDE} - q_{CDE} - \\ & q_{ABCD} + q_{ABCE} - q_{ABDE} + q_{ACDE} + q_{BCDE} - q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \bar{y}_7 = & q_0 - q_A - q_B + q_C + q_D - q_E + q_{AB} - q_{AC} - q_{AD} + q_{AE} - q_{BC} - q_{BD} + q_{BE} + q_{CD} - \\ & q_{CE} - q_{DE} + q_{ABC} + q_{ABD} - q_{ABE} - q_{ACD} + q_{ACE} + q_{ADE} - q_{BCD} + q_{BCE} + q_{BDE} - q_{CDE} + \\ & q_{ABCD} - q_{ABCE} - q_{ABDE} + q_{ACDE} + q_{BCDE} - q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_8} = & q_0 - q_A - q_B + q_C + q_D + q_E + q_{AB} - q_{AC} - q_{AD} - q_{AE} - q_{BC} - q_{BD} - q_{BE} + q_{CD} + \\ & q_{CE} + q_{DE} + q_{ABC} + q_{ABD} + q_{ABE} - q_{ACD} - q_{ACE} - q_{ADE} - q_{BCD} - q_{BCE} - q_{BDE} + q_{CDE} + \\ & q_{ABCD} + q_{ABCE} + q_{ABDE} - q_{ACDE} - q_{BCDE} + q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_9} = & q_0 - q_A + q_B - q_C - q_D - q_E - q_{AB} + q_{AC} + q_{AD} + q_{AE} - q_{BC} - q_{BD} - q_{BE} + q_{CD} + \\ & q_{CE} + q_{DE} + q_{ABC} + q_{ABD} + q_{ABE} - q_{ACD} - q_{ACE} - q_{ADE} + q_{BCD} + q_{BCE} + q_{BDE} - q_{CDE} - \\ & q_{ABCD} - q_{ABCE} - q_{ABDE} + q_{ACDE} - q_{BCDE} + q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{10}} = & q_0 - q_A + q_B - q_C - q_D + q_E - q_{AB} + q_{AC} + q_{AD} - q_{AE} - q_{BC} - q_{BD} + q_{BE} + q_{CD} - \\ & q_{CE} - q_{DE} + q_{ABC} + q_{ABD} - q_{ABE} - q_{ACD} + q_{ACE} + q_{ADE} + q_{BCD} - q_{BCE} - q_{BDE} + q_{CDE} - \\ & q_{ABCD} + q_{ABCE} + q_{ABDE} - q_{ACDE} + q_{BCDE} - q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{11}} = & q_0 - q_A + q_B - q_C + q_D - q_E - q_{AB} + q_{AC} - q_{AD} + q_{AE} - q_{BC} + q_{BD} - q_{BE} - q_{CD} + \\ & q_{CE} - q_{DE} + q_{ABC} - q_{ABD} + q_{ABE} + q_{ACD} - q_{ACE} + q_{ADE} - q_{BCD} + q_{BCE} - q_{BDE} + q_{CDE} + \\ & q_{ABCD} - q_{ABCE} + q_{ABDE} - q_{ACDE} + q_{BCDE} - q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{12}} = & q_0 - q_A + q_B - q_C + q_D + q_E - q_{AB} + q_{AC} - q_{AD} - q_{AE} - q_{BC} + q_{BD} + q_{BE} - q_{CD} - \\ & q_{CE} + q_{DE} + q_{ABC} - q_{ABD} - q_{ABE} + q_{ACD} + q_{ACE} - q_{ADE} - q_{BCD} - q_{BCE} + q_{BDE} - q_{CDE} + \\ & q_{ABCD} + q_{ABCE} - q_{ABDE} + q_{ACDE} - q_{BCDE} + q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{13}} = & q_0 - q_A + q_B + q_C - q_D - q_E - q_{AB} - q_{AC} + q_{AD} + q_{AE} + q_{BC} - q_{BD} - q_{BE} - q_{CD} - \\ & q_{CE} + q_{DE} - q_{ABC} + q_{ABD} + q_{ABE} + q_{ACD} + q_{ACE} - q_{ADE} - q_{BCD} - q_{BCE} + q_{BDE} + q_{CDE} + \\ & q_{ABCD} + q_{ABCE} - q_{ABDE} - q_{ACDE} + q_{BCDE} - q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{14}} = & q_0 - q_A + q_B + q_C - q_D + q_E - q_{AB} - q_{AC} + q_{AD} - q_{AE} + q_{BC} - q_{BD} + q_{BE} - q_{CD} + \\ & q_{CE} - q_{DE} - q_{ABC} + q_{ABD} - q_{ABE} + q_{ACD} - q_{ACE} + q_{ADE} - q_{BCD} + q_{BCE} - q_{BDE} - q_{CDE} + \\ & q_{ABCD} - q_{ABCE} + q_{ABDE} + q_{ACDE} - q_{BCDE} + q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{15}} = & q_0 - q_A + q_B + q_C + q_D - q_E - q_{AB} - q_{AC} - q_{AD} + q_{AE} + q_{BC} + q_{BD} - q_{BE} + q_{CD} - \\ & q_{CE} - q_{DE} - q_{ABC} - q_{ABD} + q_{ABE} - q_{ACD} + q_{ACE} + q_{ADE} + q_{BCD} - q_{BCE} - q_{BDE} - q_{CDE} - \\ & q_{ABCD} + q_{ABCE} + q_{ABDE} + q_{ACDE} - q_{BCDE} + q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{16}} = & q_0 - q_A + q_B + q_C + q_D + q_E - q_{AB} - q_{AC} - q_{AD} - q_{AE} + q_{BC} + q_{BD} + q_{BE} + q_{CD} + \\ & q_{CE} + q_{DE} - q_{ABC} - q_{ABD} - q_{ABE} - q_{ACD} - q_{ACE} - q_{ADE} + q_{BCD} + q_{BCE} + q_{BDE} + q_{CDE} - \\ & q_{ABCD} - q_{ABCE} - q_{ABDE} - q_{ACDE} + q_{BCDE} - q_{ABCDE} + e \end{aligned}$$

$$\overline{y_{17}} = q_0 + q_A - q_B - q_C - q_D - q_E - q_{AB} - q_{AC} - q_{AD} - q_{AE} + q_{BC} + q_{BD} + q_{BE} + q_{CD} + q_{CE} + q_{DE} + q_{ABC} + q_{ABD} + q_{ABE} + q_{ACD} + q_{ACE} + q_{ADE} - q_{BCD} - q_{BCE} - q_{BDE} - q_{CDE} - q_{ABCD} - q_{ABCE} - q_{ABDE} - q_{ACDE} + q_{BCDE} + q_{ABCDE} + e$$

$$\overline{y_{18}} = q_0 + q_A - q_B - q_C - q_D + q_E - q_{AB} - q_{AC} - q_{AD} + q_{AE} + q_{BC} + q_{BD} - q_{BE} + q_{CD} - q_{CE} - q_{DE} + q_{ABC} + q_{ABD} - q_{ABE} + q_{ACD} - q_{ACE} - q_{ADE} - q_{BCD} + q_{BCE} + q_{BDE} + q_{CDE} - q_{ABCD} + q_{ABCE} + q_{ABDE} + q_{ACDE} - q_{BCDE} - q_{ABCDE} + e$$

$$\overline{y_{19}} = q_0 + q_A - q_B - q_C + q_D - q_E - q_{AB} - q_{AC} + q_{AD} - q_{AE} + q_{BC} - q_{BD} + q_{BE} - q_{CD} + q_{CE} - q_{DE} + q_{ABC} - q_{ABD} + q_{ABE} - q_{ACD} + q_{ACE} - q_{ADE} + q_{BCD} - q_{BCE} + q_{BDE} + q_{CD} + q_{ABCD} - q_{ABCE} + q_{ABDE} + q_{ACDE} - q_{BCDE} - q_{ABCDE} + e$$

$$\overline{y_{20}} = q_0 + q_A - q_B - q_C + q_D + q_E - q_{AB} - q_{AC} + q_{AD} + q_{AE} + q_{BC} - q_{BD} - q_{BE} - q_{CD} - q_{CE} + q_{DE} + q_{ABC} - q_{ABD} - q_{ABE} - q_{ACD} - q_{ACE} + q_{ADE} + q_{BCD} + q_{BCE} - q_{BDE} - q_{CDE} + q_{ABCD} + q_{ABCE} - q_{ABDE} - q_{ACDE} + q_{BCDE} + q_{ABCDE} + e$$

$$\overline{y_{21}} = q_0 + q_A - q_B + q_C - q_D - q_E - q_{AB} + q_{AC} - q_{AD} - q_{AE} - q_{BC} + q_{BD} + q_{BE} - q_{CD} - q_{CE} + q_{DE} - q_{ABC} + q_{ABD} + q_{ABE} - q_{ACD} - q_{ACE} + q_{ADE} + q_{BCD} + q_{BCE} - q_{BDE} + q_{CDE} + q_{ABCD} + q_{ABCE} - q_{ABDE} + q_{ACDE} - q_{BCDE} - q_{ABCDE} + e$$

$$\overline{y_{22}} = q_0 + q_A - q_B + q_C - q_D + q_E - q_{AB} + q_{AC} - q_{AD} + q_{AE} - q_{BC} + q_{BD} - q_{BE} - q_{CD} + q_{CE} - q_{DE} - q_{ABC} + q_{ABD} - q_{ABE} - q_{ACD} + q_{ACE} - q_{ADE} + q_{BCD} - q_{BCE} + q_{BDE} - q_{CDE} + q_{ABCD} - q_{ABCE} + q_{ABDE} - q_{ACDE} + q_{BCDE} + q_{ABCDE} + e$$

$$\overline{y_{23}} = q_0 + q_A - q_B + q_C + q_D - q_E - q_{AB} + q_{AC} + q_{AD} - q_{AE} - q_{BC} - q_{BD} + q_{BE} + q_{CD} - q_{CE} - q_{DE} - q_{ABC} - q_{ABD} + q_{ABE} + q_{ACD} - q_{ACE} - q_{ADE} - q_{BCD} + q_{BCE} + q_{BDE} - q_{CDE} - q_{ABCD} + q_{ABCE} + q_{ABDE} - q_{ACDE} + q_{BCDE} + q_{ABCDE} + e$$

$$\overline{y_{24}} = q_0 + q_A - q_B + q_C + q_D + q_E - q_{AB} + q_{AC} + q_{AD} + q_{AE} - q_{BC} - q_{BD} - q_{BE} + q_{CD} + q_{CE} + q_{DE} - q_{ABC} - q_{ABD} - q_{ABE} + q_{ACD} + q_{ACE} + q_{ADE} - q_{BCD} - q_{BCE} - q_{BDE} + q_{CDE} - q_{ABCD} - q_{ABCE} - q_{ABDE} + q_{ACDE} - q_{BCDE} - q_{ABCDE} + e$$

$$\begin{aligned} \overline{y_{25}} = & q_0 + q_A + q_B - q_C - q_D - q_E + q_{AB} - q_{AC} - q_{AD} - q_{AE} - q_{BC} - q_{BD} - q_{BE} + q_{CD} + \\ & q_{CE} + q_{DE} - q_{ABC} - q_{AB} - q_{ABE} + q_{ACD} + q_{ACE} + q_{ADE} + q_{BCD} + q_{BCE} + q_{BDE} - q_{CDE} + \\ & q_{ABCD} + q_{ABCE} + q_{ABDE} - q_{ACDE} - q_{BCDE} - q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{26}} = & q_0 + q_A + q_B - q_C - q_D + q_E + q_{AB} - q_{AC} - q_{AD} + q_{AE} - q_{BC} - q_{BD} + q_{BE} + q_{CD} - \\ & q_{CE} - q_{DE} - q_{ABC} - q_{ABD} + q_{ABE} + q_{ACD} - q_{ACE} - q_{ADE} + q_{BCD} - q_{BCE} - q_{BDE} + q_{CDE} + \\ & q_{ABCD} - q_{ABCE} - q_{ABDE} + q_{ACDE} + q_{BCDE} + q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{27}} = & q_0 + q_A + q_B - q_C + q_D - q_E + q_{AB} - q_{AC} + q_{AD} - q_{AE} - q_{BC} + q_{BD} - q_{BE} - q_{CD} + \\ & q_{CE} - q_{DE} - q_{ABC} + q_{ABD} - q_{ABE} - q_{ACD} + q_{ACE} - q_{ADE} - q_{BCD} + q_{BCE} - q_{BDE} + q_{CDE} - \\ & q_{ABCD} + q_{ABCE} - q_{ABDE} + q_{ACDE} + q_{BCDE} + q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{28}} = & q_0 + q_A + q_B - q_C + q_D + q_E + q_{AB} - q_{AC} + q_{AD} + q_{AE} - q_{BC} + q_{BD} + q_{BE} - q_{CD} - \\ & q_{CE} + q_{DE} - q_{ABC} + q_{ABD} + q_{ABE} - q_{ACD} - q_{ACE} + q_{ADE} - q_{BCD} - q_{BCE} + q_{BDE} - q_{CDE} - \\ & q_{ABCD} - q_{ABCE} + q_{ABDE} - q_{ACDE} - q_{BCDE} - q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{29}} = & q_0 + q_A + q_B + q_C - q_D - q_E + q_{AB} + q_{AC} - q_{AD} - q_{AE} + q_{BC} - q_{BD} - q_{BE} - q_{CD} - \\ & q_{CE} + q_{DE} + q_{ABC} - q_{ABD} - q_{ABE} - q_{ACD} - q_{ACE} + q_{ADE} - q_{BCD} - q_{BCE} + q_{BDE} + q_{CDE} - \\ & q_{ABCD} - q_{ABCE} + q_{ABDE} + q_{ACDE} + q_{BCDE} + q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{30}} = & q_0 + q_A + q_B + q_C - q_D + q_E + q_{AB} + q_{AC} - q_{AD} + q_{AE} + q_{BC} - q_{BD} + q_{BE} - q_{CD} + \\ & q_{CE} - q_{DE} + q_{ABC} - q_{ABD} + q_{ABE} - q_{ACD} + q_{ACE} - q_{ADE} - q_{BCD} + q_{BCE} - q_{BDE} - q_{CDE} - \\ & q_{ABCD} + q_{ABCE} - q_{ABDE} - q_{ACDE} - q_{BCDE} - q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{31}} = & q_0 + q_A + q_B + q_C + q_D - q_E + q_{AB} + q_{AC} + q_{AD} - q_{AE} + q_{BC} + q_{BD} - q_{BE} + q_{CD} - \\ & q_{CE} - q_{DE} + q_{ABC} + q_{ABD} - q_{ABE} + q_{ACD} - q_{ACE} - q_{ADE} + q_{BCD} - q_{BCE} - q_{BDE} - q_{CDE} + \\ & q_{ABCD} - q_{ABCE} - q_{ABDE} - q_{ACDE} - q_{BCDE} - q_{ABCDE} + e \end{aligned}$$

$$\begin{aligned} \overline{y_{32}} = & q_0 + q_A + q_B + q_C + q_D + q_E + q_{AB} + q_{AC} + q_{AD} + q_{AE} + q_{BC} + q_{BD} + q_{BE} + q_{CD} + \\ & q_{CE} + q_{DE} + q_{ABC} + q_{ABD} + q_{ABE} + q_{ACD} + q_{ACE} + q_{ADE} + q_{BCD} + q_{BCE} + q_{BDE} + q_{CDE} + \\ & q_{ABCD} + q_{ABCE} + q_{ABDE} + q_{ACDE} + q_{BCDE} + q_{ABCDE} + e \end{aligned}$$

