

TRABALHO DE GRADUAÇÃO

**ANÁLISE DE MÉTRICAS DE DESEMPENHO PARA UMA REDE
AD HOC SLOTTED-ALOHA DE DOIS SALTOS
COM NETWORK CODING**

Cleyson de Vasconcelos Silva

Brasília, setembro de 2012

UNIVERSIDADE DE BRASÍLIA

FACULDADE DE TECNOLOGIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

TRABALHO DE GRADUAÇÃO

**ANÁLISE DE MÉTRICAS DE DESEMPENHO PARA UMA REDE
AD HOC SLOTTED-ALOHA DE DOIS SALTOS
COM NETWORK CODING**

Cleyson de Vasconcelos Silva

*Relatório submetido ao Departamento de Engenharia
Elétrica como requisito parcial para obtenção
do grau de Engenheiro de Redes de Comunicação*

Banca Examinadora

Prof. Renato Mariz de Moraes, ENE/UnB

Orientador

Prof. Marcelo Carvalho, ENE/UnB

Examinador interno

Prof. Anderson Clayton, ENE/UnB

Examinador interno

Dedicatória

Aos meus pais, Lisboa e Rosidete, ao meu irmão, Clenio, à minha família e meus amigos.

Cleyson de Vasconcelos Silva

Agradecimentos

Agradeço primeiramente a minha família por todo apoio e suporte que me fora dado, além da confiança em mim depositada para que pudesse passar esses anos fora de casa. Aos amigos, Chester, Goma, Aline, Mônica e suas respectivas famílias, que sempre me receberam de braços abertos e compartilharam cada momento desse processo. Aos também amigos, Heverton Rabelo, Thayron, Rodrigo, Evandro, Leandra, Alessandra, Aline Barbosa, Leidyane e Samara, que se tornaram parte de minha família também. Agradeço humildemente ao Professor Renato Mariz de Moraes por acreditar nesse projeto e sempre incitar o espírito pesquisador. Agradeço aos professores que também participaram da construção e fortalecimento desse conhecimento. Agradeço a Deus, essa força superior que nos permite enxergar a graça na simplicidade de um novo dia.

Cleyson de Vasconcelos Silva

RESUMO

O presente trabalho faz uma análise das métricas de desempenho para uma rede ad hoc Slotted Aloha e Slotted Aloha/NC(Network Coding) com dois saltos. O objetivo é descobrir a probabilidade de colisão e de slot vazio e como se comportam tais medidas com o aumento do tráfego. A análise de vazão foi realizada em [1], aqui estendemos o estudo para obter outras métricas. Nosso trabalho revelou também a existência de uma nova métrica de desempenho, um evento que está relacionado ao canal de descida no sistema. Em particular, calculamos a probabilidade do nó de repasse transmitir para os grupos de usuários, podendo assim um pacote enviado pelo nó de repasse ser codificado pela técnica de Network Coding.

ABSTRACT

This study analyzes the performance metrics of Slotted-Aloha and Slotted-Aloha/NC ad hoc networks with two hops. The objective is to discover the probability of collision and empty slots and also to verify how these metrics behave themselves with the increase of traffic. The analysis of throughput was accomplished in [1]. Here we have extended the study to achieve other metrics. This work also reveals the existence of a new metric performance, an event that is related to the downlink on the system. We evaluate the probability that the relay node transmits to group of users successfully, with or without using Network Coding.

SUMÁRIO

1	INTRODUÇÃO	1
1.1	CONTEXTUALIZAÇÃO	1
1.2	DEFINIÇÃO DO PROBLEMA	2
1.3	MOTIVAÇÃO	2
1.4	CONTRIBUIÇÃO DO TRABALHO	3
1.5	APRESENTAÇÃO DO MANUSCRITO	3
2	FUNDAMENTOS DE NETWORK CODING	4
2.1	INTRODUÇÃO	4
2.2	NETWORK CODING	4
2.3	MODELO DE REDE CABEADA	5
2.3.1	O FORMATO DO PACOTE	5
2.3.2	MODELO DE ARMAZENAMENTO	7
2.4	MODELO PARA REDES SEM FIO	8
2.4.1	COPE	9
2.4.2	GANHO DE CODIFICAÇÃO	12
2.4.3	GANHO DE CODIFICAÇÃO+MAC	12
2.4.4	CODIFICAÇÃO DE PACOTES	13
2.4.5	DECODIFICAÇÃO DE PACOTES	14
2.4.6	FORMATO DE PACOTE	14
2.4.7	CONTROLE DE FLUXO	15
3	DESENVOLVIMENTO	17
3.1	INTRODUÇÃO	17
3.2	DESCRIÇÃO DO SISTEMA	18
3.2.1	A TOPOLOGIA DA REDE	18
3.2.2	O PROTOCOLO S-ALOHA	19
3.2.3	O PROTOCOLO S-ALOHA/NC	20
3.3	ANÁLISE DAS MÉTRICAS DE DESEMPENHO	20
3.3.1	PROTOCOLO S-ALOHA	21
3.3.2	PROTOCOLO S-ALOHA/NC	24
3.4	APÊNDICES	26
3.4.1	APÊNDICE A - DEMONSTRAÇÃO DAS EQUAÇÕES (3.9) E (3.10)	26

3.4.2	APÊNDICE B - DEMONSTRAÇÃO DA EQUAÇÃO (3.24)	27
4	RESULTADOS NUMÉRICOS	29
4.1	INTRODUÇÃO	29
4.2	ANÁLISE DOS RESULTADOS	30
5	CONCLUSÃO	35
	REFERÊNCIAS BIBLIOGRÁFICAS	36
	ANEXOS	37
I	PROGRAMAS DE SIMULAÇÃO EM C	38
I.1	PROBABILIDADES DE TRANSMISSÃO DO NÓ DE REPASSE R	38
I.1.1	PROBABILIDADE DE TRANSMISSÃO $g_{b,0}$ PARA O CASO S-ALOHA	38
I.1.2	PROBABILIDADE DE TRANSMISSÃO $g_{b,0}$ PARA O CASO S-ALOHA/NC	39
I.2	MÉTRICAS DE DESEMPENHO	40
I.2.1	PROBABILIDADE DE SLOT VAZIO CASO S-ALOHA	40
I.2.2	PROBABILIDADE DE SLOT VAZIO CASO S-ALOHA/NC	42
I.2.3	PROBABILIDADES DE COLISÃO CASO S-ALOHA	43
I.2.4	PROBABILIDADES DE COLISÃO CASO S-ALOHA/NC	45
I.2.5	PROBABILIDADE DO EVENTO DE DOWNLINK CASO S-ALOHA	46
I.2.6	PROBABILIDADE DO EVENTO DE DOWNLINK CASO S-ALOHA/NC	48

LISTA DE FIGURAS

2.1	Nós A e B trocam pacotes via nó intermediário S, com e sem <i>Network Coding</i>	8
2.2	Exemplo de como <i>Network Coding</i> pode ser aplicado. Todos os enlaces tem capacidade de uma mensagem por unidade de tempo.	9
2.3	Exemplo de escuta oportuna; Nó B tem 4 pacotes em sua fila de saída, cujos próximos saltos são listados em (b). Cada vizinho de B tem armazenado alguns pacotes como mostrado em (a). O nó B pode fazer um número de decisões de codificação (como mostrado em (c)), mas selecionará a última opção por que ela maximiza o número de pacotes que são entregues em uma única transmissão. Figura retirada de [5].	11
2.4	Topologias simples para os entendimentos dos ganhos de codificação e codificação+Mac. Figura retirada de [5].	12
2.5	Estrutura do cabeçalho COPE. Figura retirada de [5].	15
2.6	Procedimentos na transmissão e recepção em COPE. Figura retirada de [5].	16
3.1	Topologia da rede. Figura retirada de [1].	18
3.2	Fila de saída e Camada Mac para os protocolos S-ALOHA e S-ALOHA/NC. Figura retirada de [1].	19
3.3	Cadeia de Markov para os estados V_k da fila de saída do nó R. Figura retirada de [1].	21
3.4	Cadeia de Markov para os estados W_k da fila de saída do nó R. Figura retirada de [1].	24
4.1	Comparação entre as probabilidades de transmissão do nó R.	31
4.2	Probabilidades de colisão e slot vazio para o grupo 1.	31
4.3	Probabilidade do nó R transmitir um pacote no subsistema formado pelo grupo 1 e o nó R.	32
4.4	Probabilidades de colisão e slot vazio para o grupo 2.	32
4.5	Probabilidade do nó R transmitir um pacote no subsistema formado pelo grupo 2 e o nó R.	33
4.6	Curvas de vazão.	33

LISTA DE TABELAS

2.1	Tabela de definições dos termos usados em COPE.....	10
2.2	Ganhos teóricos para topologias básicas	13

LISTA DE SÍMBOLOS

Símbolos Latinos

e	Aresta de um grafo
E	Número de arestas em um grafo
G	Grafo orientado
G_v	Tráfego gerado pelo grupo v
g_v	Tráfego gerado por usuário no grupo v
g_0	Tráfego do nó de Repasse
$g_{b,0}$	Probabilidade de transmissão do nó de Repasse
G_t	Matriz de decodificação
$g(e)$	Vetor de codificação global
h	Máxima taxa de informação enviada pela fonte de acordo com o Teorema do Máximo fluxo Mínimo corte
$m(e)$	Vetor de codificação local
m_v	Número de usuários no grupo v
n_v	Número de pacotes na fila virtual do grupo v
P_D	Probabilidade de decodificar um pacote codificado
$R_{i,j}$	Capacidade do enlace i - j
s	Fonte
S	Vazão total
S_v	Vazão do grupo v
T	Nós destinos
V	Número de vértices em um grafo
x	Símbolo da fonte
$y(e)$	Símbolo combinado

Símbolos Gregos

η_v	Probabilidade de não haver transmissão no grupo
γ_v	Probabilidade de um grupo transmitir com sucesso
λ	Probabilidade de transição de chegada na cadeia de markov
μ	Probabilidade de transição de partida na cadeia de markov
ρ	Fator de utilização
$\varphi(.)$	Probabilidade estacionária para o caso S-Aloha
ξ_v	Probabilidade de haver slot vazio para o grupo v
σ_v	Probabilidade de haver colisão no grupo v
ϱ_v	Probabilidade de haver o evento de downlink no grupo v
$\psi(.)$	Probabilidade estacionária para o caso S-Aloha/NC
θ	Indicador de maior probabilidade de transmissão entre os grupos

Capítulo 1

Introdução

Este capítulo apresenta os aspectos principais do trabalho, contextualizando o problema proposto no âmbito das redes ad hoc sem fio. Além disso, a motivação e as contribuições são destacadas, e a estrutura do manuscrito é apresentada.

1.1 Contextualização

Redes *ad hoc* [2] são caracterizadas por serem descentralizadas, ou seja, não há uma entidade que controla os recursos da rede, os próprios nós o fazem. Roteamento, armazenamento, sinalização e outros recursos são administrados sem a necessidade de um nó que gerencie os demais. Pelo fato de ser uma rede sem infraestrutura, a comunicação pode ser realizada com um único salto ou múltiplos saltos, e pode variar a todo momento por questões de mobilidade, queda de enlace e outros eventos. Como a informação pode ser propagada através de vários saltos por nós intermediários, estes são de suma importância para o processo, pois a capacidade de um sistema pode ser reforçada substituindo um enlace de baixa qualidade e longa distância por vários de curta distância e alta qualidade. O protocolo de comunicação chamado *Aloha* [3] é um protocolo de controle de acesso ao meio baseado em contenção no qual não existe uma ordem de acesso, e nada impede que haja colisão acarretando perda de mensagens, pois tal protocolo não detecta ocupação do canal, ou seja, um nó pode transmitir mesmo o canal estando em uso. A detecção de colisão é feita pelo disparo de um temporizador aleatório durante a transmissão. Caso um quadro de reconhecimento de transmissão não chegue antes do temporizador expirar, o quadro original é retransmitido. A aleatoriedade do temporizador existe para reduzir a probabilidade de novas colisões. Esse protocolo fornece uma vazão de dados muito baixa, com altas taxas de colisão e ociosidade da rede. O protocolo que se seguiu foi uma aprimoração do *Aloha*, e foi denominado de *Slotted Aloha*. A única mudança introduzida é que agora os nós estarão sincronizados para transmitir (ou não) em momentos específicos. O tempo é dividido em espaços regulares do tamanho de um quadro. Assim as estações sincronizadas sabem exatamente onde os intervalos começam. Dessa forma, os nós só podem transmitir no início desse intervalo. O aperfeiçoamento do protocolo consiste em reduzir o tempo de colisão fazendo com que os pacotes colidam por inteiro. Qualquer colisão terá o tempo de um quadro. Apesar de dobrar a vazão da rede, o desempenho do protocolo é muito baixo ainda.

Contudo é um bom ganho dada a simplicidade do sistema.

Muitas são as técnicas incorporadas aos protocolos de comunicação para melhorar o desempenho do sistema na tarefa de acesso ao meio. Uma área de pesquisa recente na Teoria da Informação é a técnica de codificação de pacotes (do inglês *Network Coding* (NC) [4]. Realizando combinação de pacotes, um nó pode transmitir a informação de vários pacotes em um só, aumentando dessa forma a vazão e reduzindo o atraso na rede. A codificação pode ser obtida realizando uma combinação linear dos pacotes que se deseja enviar a partir dos coeficientes de um corpo estendido de Galois, onde esses coeficientes podem ser gerados aleatoriamente, ou simplesmente obter um pacote combinado realizando a operação *ou exclusivo* (XOR, do inglês *exclusive or*) sobre os demais. Com esta técnica, redes baseadas em difusão tem maior vantagem, uma vez que o meio é compartilhado. Assim podendo enviar um maior número de pacotes aos receptores com somente uma transmissão. Concomitantemente o fato de haverem nós de repasse na rede, principalmente em redes *ad hoc*, faz com que as opções de codificação na mesma sejam maiores melhorando seu desempenho. As aplicações são diversas, redes veiculares, monitoramento de atividades sísmicas, mapeamento de regiões através de sensores ou qualquer outra que não dependa de centralização.

1.2 Definição do problema

O presente trabalho faz um estudo em redes *ad hoc* com dois saltos. Dois grupos de usuários trocam pacotes de informação através de um nó intermediário, e todos os nós do sistema são estáticos. O objetivo principal é comparar as métricas de desempenho de colisão e ociosidade para esse sistema operando com o protocolo de acesso ao meio Slotted-Aloha e Slotted-Aloha/NC, uma vez que a vazão foi calculada em [1]. Este último adiciona a técnica de codificação baseada em combinação de pacotes chamada de *Network Coding* que será explicada nos capítulos posteriores. Entender o comportamento dessas métricas se faz essencial para a melhora do sistema. A grande questão é que no cenário estabelecido para estudo apenas o nó intermediário faz *network coding*. Quebrando assim a limitação de apenas repassar a informação, realizar combinação de pacotes aumenta a capacidade da rede sem alterar os recursos disponibilizados pela mesma, sem falar de questões de segurança, tópico que não será abordado nesse trabalho mas que também é uma forte área de estudo dentro de *Network Coding*. A análise aqui descrita se faz em caracterizar os canais entre cada grupo e o nó intermediário, mapeando os eventos que podem ocorrer e agrupando-os em suas categorias, colisão ou slot vazio.

1.3 Motivação

É de suma importância compreender todas as métricas de desempenho de um sistema, muitos trabalhos direcionam seus estudos apenas a vazão, deixando de lado a taxa de colisão e ociosidade do sistema. Obter os valores dessas métricas para níveis distintos de intensidade de tráfego pode ajudar a melhorar a eficiência do sistema, não só em relação a vazão mas também em economia

de energia, se um dado nó de uma rede de sensores souber a probabilidade do sistema ficar ocioso para determinado nível de tráfego o nó pode ficar inativo para economizar energia.

1.4 Contribuição do trabalho

O trabalho se concentra no estudo das métricas de desempenho e caracterização do canal afim de entender o comportamento do sistema como um todo. Obter as probabilidades de colisão e slot vazio e seu comportamento com a variação do tráfego gerado pelos grupos de usuários, dessa forma mapeando cada evento no canal de comunicação. Em nossa análise foi descoberto um evento que não se enquadra em nenhuma das outras métricas, este relaciona-se ao canal de descida do sistema, sendo um ponto de suma importância para avaliação e comparação dos resultados aqui descritos. Estendemos a análise das métricas para uma maior faixa de tráfego na rede, verificando seu comportamento em situações de congestionamento e baixa utilização.

1.5 Apresentação do manuscrito

No Capítulo 2 são explicadas maneiras de se realizar *network coding*, dentre elas a técnica COPE [5], que é o método de codificação utilizado neste trabalho, bastante conveniente para a topologia de rede em questão, devido a sua simplicidade. Em seguida, o Capítulo 3 descreve com detalhes a topologia do sistema e o desenvolvimento analítico para as métricas de desempenho. Os resultados numéricos são apresentados no Capítulo 4 seguido das conclusões finais no Capítulo 5 e dos códigos em linguagem C nos anexos.

Capítulo 2

Fundamentos de Network Coding

Este capítulo apresenta maneiras de se realizar network coding em redes ponto-a-ponto e sem fio, estrutura do pacote, codificação e decodificação.

2.1 Introdução

Neste capítulo serão abordados conceitos fundamentais para a realização de *Network Coding*, tal como os processos de codificação e decodificação de pacotes. Dependendo do cenário de rede, a codificação pode ser realizada de maneiras distintas, mas mantendo a filosofia de combinar pacotes. Em [6], *Ahlsvede et al* descreve um ambiente de rede de comunicação ponto-a-ponto com um único nó fonte que realiza transmissões multicast para nós específicos através de nós intermediários que são capazes de combinar pacotes através de uma rede física, e em [5], é feita uma abordagem teórica de como usar *Network Coding* em uma rede ad hoc, especificando uma nova arquitetura de codificação inserida entre a camada de acesso ao meio (MAC, do inglês medium access control) e a camada de rede. As seções seguintes explicam com mais detalhes essas técnicas.

2.2 Network Coding

Network Coding é um campo de estudo muito recente, que teve seu "nascimento"ano 2000 com o trabalho de *Ahlsvede et al.* [6]. A definição mais comum de *Network Coding* é a de que um nó dentro da rede é capaz de fazer codificação: nós de repasse agora não somente duplicam e encaminham pacotes, eles os combinam. Com *Network Coding* os nós podem propiciar um tratamento mais sofisticado aos pacotes em suas filas de saída, tratamento esse que permite aumentar a vazão da rede tornando-a também robusta à perda de pacotes e enlaces com falhas. É uma técnica que oferece segurança e confiabilidade, pois um invasor na rede que consegue receber alguns pacotes codificados pela técnica de *Network Coding* dificilmente conseguirá decodificá-los. A codificação pode ser realizada de diversas formas, não importando se a rede é infraestruturada ou do tipo ad hoc, fixa ou móvel. Em [7], [4] são mostrados com mais detalhes e profundidade matemática os modelos aqui apresentados, visto que um aprofundamento algébrico foge do escopo desse trabalho. Contudo a idéia se matém bem clara e o tratamento matemático dado nas seções seguintes é o

suficiente para nossos objetivos.

Uma importante propriedade de operação de rede com *Network Coding* é que, para alguns cenários de tráfego, a codificação permite efetivamente que os nós da rede alcancem um ótimo desempenho quando atuando em uma configuração descentralizada. Isto encontra aplicações imediatas em ambientes com alterações dinâmicas, onde o controle e gerenciamento de redes centralizadas tem excessiva complexidade. Assim, os nós da rede precisam operar de uma maneira distribuída sem conhecimento global da configuração da rede. Tais aplicações ocorrem em redes de distribuição ponto-a-ponto, e nas redes sem fio onde a configuração de rede pode se alterar devido a mobilidade dos nós. A seguir são descritas duas formas de se realizar *Network Coding*.

2.3 Modelo de Rede Cabeada

2.3.1 O Formato do Pacote

Consideremos então um grafo acíclico direcionado $G=(V,E)$, onde V é número de vértices e E o conjunto de arestas desse grafo. A informação percorre sem ruído a partir um nó i para o nó j passando pelo caminho $(i,j) \in E$. Um exemplo desse tipo de rede é um backbone. Para este modelo, o grafo pode ser representado como um rede física, enquanto os dados multicast podem ser o tráfego padrão agregado que o sistema deve suportar, ou ainda G pode ser uma sub-rede dentro da rede física e os dados multicast podem pertencer a uma aplicação específica, como por exemplo, uma chamada de vídeo-conferência. Seja $s \in V$ a fonte que transmite a uma dada taxa de informação h (medida em bits por segundo), para um conjunto de nós destinos $T=(t_1, t_2, \dots, t_m) \in V$, $R_{i,j}$ a capacidade do enlace $e \in E$. Assumindo que o fluxo total de informação que entra no nó i é igual ao fluxo total que sai desse nó, temos que, pelo **Teorema do Máximo Fluxo Mínimo Corte** [8] a taxa de informação máxima que deve ser enviada por s para os destinatários é igual ao menor corte na rede, sendo o corte uma partição que separa a fonte do destino. Assumimos também que o fluxo de símbolos está sincronizado em toda a rede.

Sabendo a topologia do grafo, é possível realizar codificação da maneira descrita a seguir. A fonte s deseja enviar informação comum ao conjunto de nós receptores $T \subset V$ a uma taxa h . Cada laço e saindo do nó i carrega um símbolo $y(e)$ que é uma combinação linear dos símbolos $y(e')$ que chegam em i pelos laços e' . Tal combinação pode ser expressa como $y(e) = \sum_{e'} m_e(e')y(e')$, onde $\mathbf{m}(e) = [m_e(e')]$ é chamado de *vetor de codificação local* e representa a função de codificação do nó i para o laço e . Caso i seja a fonte s , a fim de manter-se a notação uniforme, laços virtuais e'_1, e'_2, \dots, e'_h são inseridos em s carregando os h símbolos fonte $y(e'_k)=x_k, k=1,2,\dots,h$. Assim, por indução, algum $y(e)$ sobre um laço $e \in E$ é representado pela combinação linear $y(e) = \sum_{k=1}^h g_k(e)x_k$ de símbolos fonte, onde $\mathbf{g}(e) = [g_1(e), g_2(e), \dots, g_h(e)]$ é um vetor de dimensão h chamado de *vetor de codificação global*, e pode ser determinado recursivamente por $\mathbf{g}(e) = \sum_{e'} m_e(e')\mathbf{g}(e')$, onde $\mathbf{g}(e'_k)$ é o resultado das combinações lineares que os símbolos da fonte sofreram até esse ponto da rede. Dessa forma um dado receptor t , recebendo os h (ou mais) símbolos $y(e_1), \dots, y(e_h)$, pode obter os símbolos

fonte a partir da matriz de codificação G_t da seguinte forma:

$$\begin{bmatrix} y(e_1) \\ \vdots \\ y(e_h) \end{bmatrix} = \begin{bmatrix} g_1(e_1) & \cdots & g_h(e_1) \\ \vdots & \cdots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix} = G_t \begin{bmatrix} x_1 \\ \vdots \\ x_h \end{bmatrix}$$

G_t é a matriz de codificação global de ordem h composta pelos vetores $\mathbf{g}(e_1), \dots, \mathbf{g}(e_h)$. A decodificação irá acontecer com alta probabilidade se os vetores de codificação forem gerados aleatoriamente e os símbolos pertencerem a um corpo finito de tamanho suficiente, ou seja, os símbolos devem pertencer a uma extensão do corpo algébrico Z_2 . De acordo com [9], se o tamanho do campo é da ordem de 2^{16} e o número de enlaces na rede for $|\mathbf{E}|=2^8$, então a matriz G_t em um dado receptor será quadrada de ordem h com probabilidade de no mínimo $1 - 2^{-8} = 0,996$.

Em uma rede de pacotes, os símbolos $y(e)$ transportados pelo laço e podem ser agrupados em pacotes. Na Internet o tamanho máximo de um típico pacote, excluindo seu cabeçalho, é da ordem de 1400 bytes. Assim cada pacote IP pode carregar $N=1400$ símbolos se estes pertencem a um corpo da ordem de 2^8 ou $N=700$ símbolos se a ordem do corpo algébrico for 2^{16} . Desse modo empacotam-se os símbolos $y(e)$ fluindo sobre o laço e em vetores $\mathbf{y}(e) = [y_1(e), y_2(e), \dots, y_N(e)]$ de tamanho apropriado, dependendo do tamanho do corpo utilizado. Assim cada um desses vetores pode ser expresso como uma combinação linear $\mathbf{y}(e) = \sum_{e'} m_e(e') \mathbf{y}(e')$ de vetores $\mathbf{y}(e')$ entrando em i pelos laços e' . Os símbolos fonte são empacotados em vetores $\mathbf{x}_k = [x_{k,1}, x_{k,2}, \dots, x_{k,N}]$, logo um receptor pode decodificar (com alta probabilidade) os h vetores fonte $\mathbf{x}_1, \dots, \mathbf{x}_h$ a partir de quaisquer h pacotes recebidos, isto é,

$$\begin{bmatrix} \mathbf{y}(e_1) \\ \vdots \\ \mathbf{y}(e_h) \end{bmatrix} = \begin{bmatrix} y_1(e_1) & y_2(e_1) & \cdots & y_N(e_1) \\ \vdots & \vdots & \ddots & \vdots \\ y_1(e_h) & y_2(e_h) & \cdots & y_N(e_h) \end{bmatrix} = G_t \begin{bmatrix} \mathbf{x}_1 \\ \vdots \\ \mathbf{x}_h \end{bmatrix} = G_t \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ x_{h,1} & x_{h,2} & \cdots & x_{h,N} \end{bmatrix}$$

É ainda preciso incluir dentro de cada pacote fluindo sobre o laço e o vetor de codificação global $\mathbf{g}(e)$ de dimensão h . Com isso um receptor pode obter os pacotes fontes, pois detém os pacotes codificados e os vetores de codificação globais referentes a tais pacotes, dessa forma é suficiente gerar a inversa da matriz G_t e aplicar eliminação de Gauss. Logo,

$$\begin{bmatrix} g_1(e_1) & \cdots & g_h(e_1) & y_1(e_1) & \cdots & y_N(e_1) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ g_1(e_h) & \cdots & g_h(e_h) & y_1(e_h) & \cdots & y_N(e_h) \end{bmatrix} = G_t \begin{bmatrix} 1 & \cdots & 0 & x_{1,1} & \cdots & x_{1,N} \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & 1 & x_{h,1} & \cdots & x_{h,N} \end{bmatrix}$$

O custo desse esquema é a inclusão dos h símbolos, representados pelo vetor de codificação global, em cada pacote. Se h for 50, tem-se um acréscimo de aproximadamente 3% do tamanho do pacote, para pacotes que tenham tamanho de 1400 bytes que pertençam a um corpo de ordem 2^8 , o que é razoável em troca dos benefícios que tal codificação traz. Os receptores não precisam conhecer das

funções de codificação e topologia da rede, podem decodificar mesmo se enlaces forem adicionados ou removidos, como em um cenário *ad hoc*, e também mesmo que enlaces falhem e haja perda de pacotes. Outra característica, é que se pode decodificar pacotes mesmo se os vetores de codificação local sejam variantes no tempo e randomicamente escolhidos. Contudo, apesar da eficiência desse formato de pacote, é preciso ainda um modelo de armazenamento para as filas de saída para se saber quais pacotes se deve combinar.

2.3.2 Modelo de armazenamento

O esquema para o formato de pacote não satisfaz, por si só, as necessidades para redes reais. Pacotes em diferentes enlaces estão sujeitos a diferentes propagações e atrasos de filas, e o número de pacotes associados ao mesmo conjunto de vetores fontes pode variar por causa da perda de pacotes, congestionamento, ou mudanças relacionadas à largura de banda disponível devido à disputa pelo tráfego. Por essas razões, torna-se extremamente importante a sincronização dos pacotes relacionados ao mesmo conjunto de pacotes fonte $\mathbf{x}_1, \dots, \mathbf{x}_h$, tanto na codificação quanto na decodificação. Dessa forma vamos considerar que todos os pacotes associados ao conjunto de h vetores fonte $\mathbf{x}_1, \dots, \mathbf{x}_h$, sejam chamados de pacotes da mesma *geração*, e h é chamado de *tamanho da geração*. Todos os pacotes da mesma geração são marcados com o mesmo *número de geração*. As gerações recebem números sequenciais, um ou dois bytes no cabeçalho do pacote são suficientes para distinguir entre as sucessivas gerações em uma rede.

Além da adição do *número de geração* no cabeçalho, é preciso um mecanismo de sincronização para a chegada e partida de pacotes. Nesse modelo, os pacotes que chegam no nó por quaisquer laços são armazenados em uma única fila (*buffer*) de acordo com seu número de geração. Dessa maneira, quando há uma oportunidade de transmissão, um pacote é gerado contendo uma combinação linear aleatória de todos os pacotes que já se encontram dentro do buffer da atual geração. Periodicamente todos os pacotes de uma geração são descartados do buffer dando lugar à geração com número de geração seguinte. Pacotes que chegam marcados com números de geração anteriores são descartados. Os pacotes que chegam no buffer para uma dada geração aumenta em número com o decorrer do tempo, e uma combinação linear é escolhida aleatoriamente para cada pacote de saída, assim as funções de codificação são variantes no tempo. No entanto isso não é problema, pois como visto na subsecção anterior, os vetores de codificação global são inseridos no pacote.

Uma outra medida adotada é referente aos pacotes que chegam no buffer e têm o mesmo vetor de codificação global \mathbf{g} . Tais pacotes são chamados *não inovadores*, pois o vetor \mathbf{g} que ele carrega já pertence ao subespaço de vetores de codificação global oriundo dos pacotes recebidos. Logo tal pacote nada acrescenta ao processo de decodificação, sendo assim descartado. Porém, o pacote que chega com um vetor \mathbf{g} que não pertence ao subespaço citado é dito *inovador*. Assim a informação não inovadora não atrapalha a transmissão da informação inovadora, com isso um nó pode transmitir livremente sobre todos os seus enlaces de saída sem ter o conhecimento de que a informação que ele transmite será inovadora ou não para seus vizinhos. Pacotes não inovadores consomem

largura de banda que poderia ser usada para outros propósitos, contudo em uma configuração distribuída é possível salvar largura de banda, desde que os nós monitorem as taxas de pacotes inovadores e compartilhem com seus vizinhos para restringir a transmissão apenas para pacotes da geração atual. Uma outra alternativa para economizar recursos de rede é rodar um algoritmo de máximo fluxo entre o emissor e os receptores e então restringir a codificação de rede a partir desses fluxos.

2.4 Modelo para Redes sem Fio

O modelo descrito nessa subsecção é chamado COPE [5], que é uma arquitetura para *Network Coding* em redes sem fio. COPE insere uma camada de codificação entre a camada de rede e a camada de enlace, que identifica oportunidades de codificação e benefícios para a transmissão de múltiplos pacotes em uma única transmissão. O exemplo mais simples de codificação com COPE está mostrado na Fig.2.1, onde dois nós trocam pacotes através de um nó intermediário.

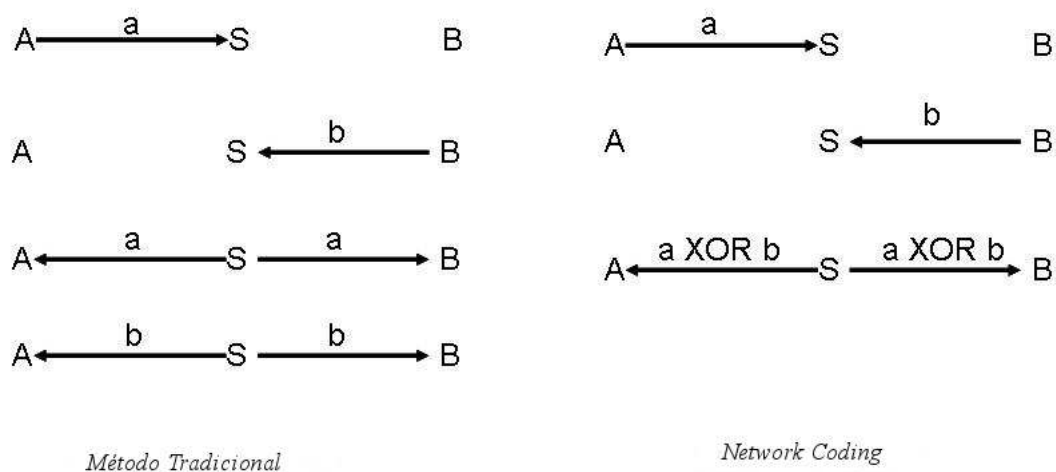


Figura 2.1: Nós A e B trocam pacotes via nó intermediário S, com e sem *Network Coding*.

Percebe-se claramente que no método tradicional se faz necessário quatro envios para que a informação seja repassada para os destinos, enquanto que com *Network Coding* ambos os nós decodificam a informação com apenas 3 transmissões. O uso de *Network Coding* em sistemas de radiodifusão é muito eficiente, uma vez que o meio é compartilhado por todos os nós da rede. Logo um nó pode enviar vários pacotes para nós distintos em uma única transmissão.

O exemplo mais famoso de *Network Coding* é uma topologia de rede chamada de borboleta (*butterfly*), e a diferença para o exemplo da Fig.2.1 é que o meio não é compartilhado, porém o mesmo objetivo é alcançado. Graficamente, temos que na Fig.2.2 a fonte S_1 deseja entregar a_i para R_1 e R_2 , e a fonte S_2 deseja entregar b_i para ambos receptores também. O nó intermediário é tido como um gargalo na rede, visto que se não fizer codificação, no tempo de uma transmissão, ele enviará a_i para R_2 ou b_i para R_1 . Se houver codificação de rede no nó intermediário, ambos

os receptores recebem as mensagens na mesma unidade de tempo. Para tráfego multicast, como na Fig.2.2, códigos lineares aleatórios são suficientes para alcançar a máxima capacidade da rede, como os vetores de codificação gerados pelas funções aleatórias mostrados na subsecção anterior. Para explorarmos com mais detalhes COPE, é apresentada a Tabela 2.1 com definições importantes para o entendimento da metodologia de codificação de pacotes para redes wireless.

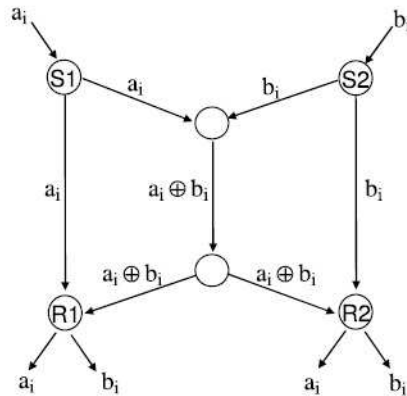


Figura 2.2: Exemplo de como *Network Coding* pode ser aplicado. Todos os enlaces tem capacidade de uma mensagem por unidade de tempo.

2.4.1 COPE

COPE é uma nova arquitetura para redes descentralizadas que insere uma camada de codificação entre as camadas de roteamento e enlace, detectando oportunidades de codificação e explorando-as para enviar múltiplos pacotes em uma única transmissão. Com as definições da Tabela 2.1 é possível entender as técnicas incorporadas por COPE. São elas:

1. Escuta Oportuna: Em redes sem fio, os usuários compartilham o meio. Dessa forma criam-se algumas oportunidades para que os nós, equipados com antenas omni-direcionais, possam ouvir os pacotes. COPE inicia os nós em modo promíscuo fazendo que todos se comuniquem e armazenem pacotes por um período de T segundos, por padrão, $T=0.5s$. Cada nó do meio envia *informes de recepção* para seus vizinhos, informando-os dos pacotes que tem armazenado durante esse tempo. Os informes de recepção são enviados dentro do pacote, caso um nó não tenha pacotes para transmitir, ele envia os informes em pacotes de controle.
2. Oportunidade de Codificação: O cerne da questão é que, pacotes codificados juntos (combinados), maximizam a capacidade da rede. Um nó pode ter inúmeras possibilidades de codificação, mas escolherá aquela que maximiza o número de pacotes nativos entregues em uma única

Tabela 2.1: Tabela de definições dos termos usados em COPE.

Termos	Definições
Pacote nativo	Um pacote não codificado
Pacote codificado	Pacote gerado pela combinação de vários pacotes nativos através da operação XOR
Proximos saltos de um pacote codificado	Conjunto de nós a que será enviado o pacote codificado
Id do pacote	Um hash de 32 bits do endereço IP da fonte e o número de sequência IP do pacote
Fila de saída	Uma fila FIFO em cada nó, onde mantém os pacotes que precisa encaminhar
Reservatório de pacotes (packet pool)	Um buffer onde um nó armazena todos os pacotes ouvidos em um período anterior de T segundos
Ganho de codificação	A razão entre o número de transmissões necessárias para uma abordagem sem codificação e o número de transmissões usadas por COPE para entregar o mesmo conjunto de pacotes
Ganho de codificação +Mac	É o ganho de capacidade esperado com COPE quando o padrão 802.11 Mac é utilizado e todos os nós tem tráfego a oferecer

transmissão, assegurando que cada próximo salto pretendido tenha informação suficiente para decodificar o pacote nativo enviado a ele. A Fig.2.3 ilustra bem esse exemplo.

O nó B tem 4 pacotes, p_1 , p_2 , p_3 e p_4 . Seus vizinhos ouviram alguns desses pacotes. A tabela na Fig.2.3(b) mostra o próximo salto para cada pacote na fila de B. Quando o método permite que B transmita, B usa o pacote p_1 do começo da fila. Assumindo que B tem conhecimento dos pacotes que cada vizinho possui, algumas opções de codificação são levantadas como mostrado na Fig.2.3(c). Ele poderia enviar $p_1 + p_2$. Assim, C tem p_1 armazenado, se realizar um XOR de p_1 com $p_1 + p_2$, poderá obter p_2 que é o pacote nativo destinado a ele. Entretanto, o nó A não possui p_2 , logo não pode decodificar o pacote combinado. Dessa forma a combinação proposta seria uma má decisão de codificação para B, pois apenas um dos vizinhos se beneficiaria com a transmissão. A segunda opção na Fig.2.3(c) mostra uma melhor decisão de codificação para B. Enviando $p_1 + p_3$ permitiria a ambos os vizinhos A e C decodificar e obter os pacotes nativos destinados a eles a partir de uma única transmissão. Porém a melhor decisão de codificação seria enviar $p_1 + p_2 + p_3$, pois essa decisão permitiria que todos os três vizinhos recebam seus respectivos pacotes de uma só vez.

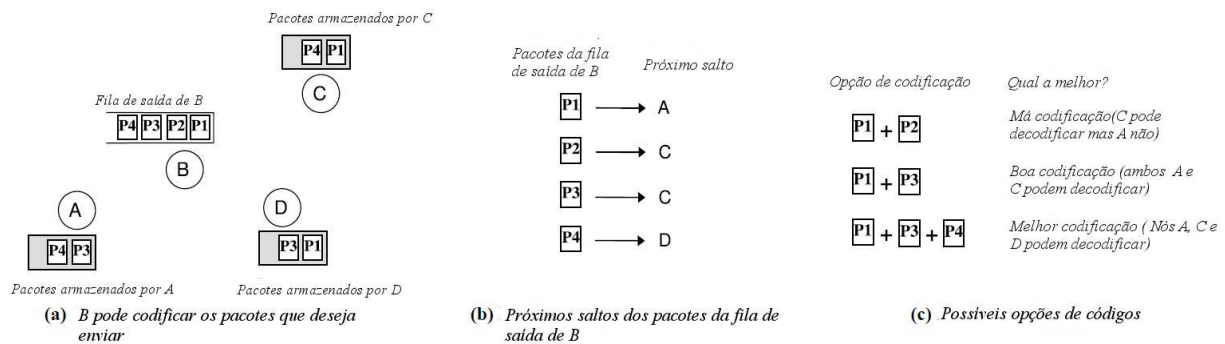


Figura 2.3: Exemplo de escuta oportuna; Nó B tem 4 pacotes em sua fila de saída, cujos próximos saltos são listados em (b). Cada vizinho de B tem armazenado alguns pacotes como mostrado em (a). O nó B pode fazer um número de decisões de codificação (como mostrado em (c)), mas selecionará a última opção por que ela maximiza o número de pacotes que são entregues em uma única transmissão. Figura retirada de [5].

O exemplo acima aborda um importante problema de codificação. Pacotes vindos de fontes distintas podem ser codificados em um mesmo nó intermediário, e para que não haja perda de capacidade, ou seja, um determinado destino não consiga decodificar o pacote combinado e obter o pacote nativo destinado a ele, o algoritmo de codificação deve assegurar que todos os próximos saltos de um pacote codificado possam decodificar seus correspondentes pacotes nativos. Para isso, tem-se a seguinte regra: *Para transmitir n pacotes, p_1, \dots, p_n , para n vizinhos, r_1, \dots, r_n , o nó pode combinar os n pacotes juntos, somente se cada próximo salto r_i tiver todos os $n-1$ pacotes p_j para i diferente de j .* Esta regra assegura que cada próximo salto pode decodificar o pacote combinado e extrair o pacote nativo a ele destinado. Então, quando um nó tem a chance de transmitir um pacote, ele escolhe o maior n que satisfaz a regra para maximizar o benefício da codificação.

3. Aprendendo o estado do vizinho: Como um nó sabe quais pacotes um dado vizinho possui? Como visto anteriormente, os nós trocam informes de recepção anunciando à seus vizinhos os pacotes que tem recebido. Entretanto, em tempos de grandes congestionamentos os informes de recepção podem ser perdidos em colisões, ou podem chegar atrasados, depois que o nó já realizou uma decisão de codificação. Contudo, um nó não pode depender somente dos informes, ele pode precisar adivinhar se um dado vizinho tem um pacote em particular. Para adivinhar de forma inteligente, é levantado um cálculo de roteamento. Protocolos de roteamento sem fio calculam a probabilidade de entrega entre cada par de nós e as usam para identificar os melhores caminhos. Ocasionalmente, um nó pode adivinhar de maneira errônea, e isso pode fazer com que o pacote codificado se torne não-decodificável para algum próximo salto. Neste caso, o pacote nativo destinado a esse vizinho será combinado novamente com um novo conjunto de pacotes nativos.

2.4.2 Ganho de Codificação

O *ganho de codificação* é a razão entre o número de transmissões necessárias pelo método tradicional e o número de transmissões requeridas por COPE para enviar o mesmo conjunto de pacotes. A Fig.2.1 ilustra muito bem esse ganho, o uso de network coding reduz o número de transmissões de 4 para 3, resultando em um ganho de codificação de $\frac{4}{3} = 1.33$. COPE admite nós idênticos, com antenas omnidirecionais com perfeita escuta no raio de transmissão.

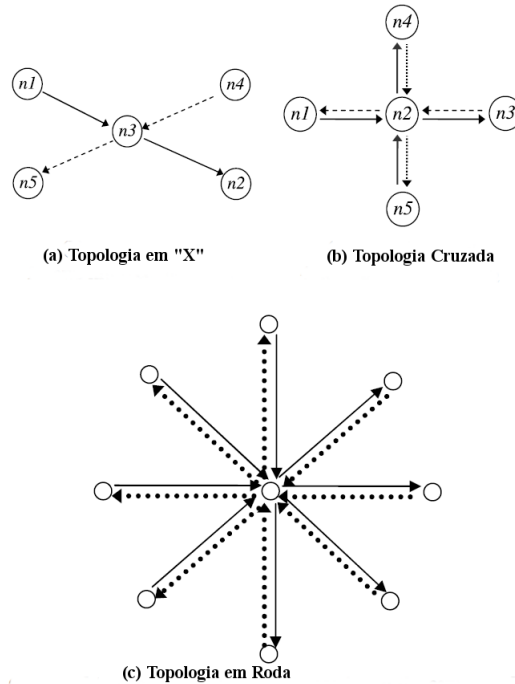


Figura 2.4: Topologias simples para os entendimentos dos ganhos de codificação e codificação+Mac. Figura retirada de [5].

A Fig.2.4 mostra algumas topologias para explicar os ganhos de codificação. Na topologia "X" na Fig.2.4(a) tem-se também um ganho de 1.33 como na Fig.2.1, porém na Fig.2.4(b) percebe-se que são necessárias 8 transmissões para que todos os nós possam receber os pacotes destinados, sendo que com *Network Coding* bastam 5 transmissões. Todo nó com exceção do nó de repasse só não escuta o nó diametralmente oposto, isso acarreta um ganho de $\frac{8}{5} = 1.6$. Com esse mesmo raciocínio na Fig.2.4(c) obtemos um ganho de codificação de $\frac{16}{9} = 1.77$, quanto mais nós ao alcance do nó de repasse, maior o ganho.

2.4.3 Ganho de Codificação+Mac

O ganho de codificação+Mac está relacionado com o número de pacotes drenados da fila de saída do nó de repasse. Na Fig.2.1 o nó intermediário consegue enviar dois pacotes em uma única transmissão, obtendo um ganho +Mac de 2. Observando as topologias em Fig.2.4(b) e Fig.2.4(c) temos ganhos de 4 e 8 respectivamente. O ganho de codificação+Mac expressa quantos pacotes podem ser combinados e enviados em uma única transmissão com a garantia de que possam ser

decodificados por seus destinos. A tabela 2.2 resume os ganhos para as topologias apresentadas na Fig.2.4.

Tabela 2.2: Ganhos teóricos para topologias básicas

Topologia	Ganho de Codificação	Ganho de Codificação+Mac
"X"	1.33	2
Cruzada	1.6	4
Roda	1.77	8

2.4.4 Codificação de pacotes

Para a construção do esquema de codificação algumas decisões de projeto devem ser tomadas. A primeira delas se baseia em *nunca atrasar os pacotes*, ou seja, quando o canal sem fio esta disponível, o nó usa o primeiro pacote da sua fila de saída e verifica quais outros pacotes na sua fila podem ser combinados com esse. Então gera um pacote codificado a partir dos pacotes nativos, e se não há oportunidades de codificação, o nó não espera pela chegadas de pacotes para que possa combinar, ele envia o pacote nativo.

Segundo, COPE *dá preferência à combinação de pacotes de tamanhos similares*, pois combinar pacotes de tamanhos distintos reduz a largura de banda salva. Nesse tipo de combinação, os pacotes menores são completados com zeros. Os receptores podem remover facilmente os bits acrescentados checando o campo de tamanho do pacote no cabeçalho IP de cada pacote nativo.

Terceiro, *nunca codificar pacotes que tenham como destino o mesmo próximo salto*, pois dessa forma o nó que representa o próximo salto citado não conseguirá decodificá-lo. COPE mantém duas filas virtuais por vizinho, uma para pacotes pequenos e outra para pacotes grandes. Quando um pacote é adicionado à fila de saída, um ponteiro é adicionado na fila virtual apropriada, baseado no tamanho e próximo salto do pacote. A busca por pacotes apropriados para codificar é eficiente devido as filas virtuais. Quando vai tomar uma decisão de codificação, COPE primeiro retira o primeiro pacote da fila de saída e determina se é um pacote pequeno ou não, e dependendo do tamanho, ele verifica as filas virtuais adequadas. Se for um pacote pequeno, por exemplo, verifica as filas virtuais para pacotes pequenos, apenas o primeiro de cada fila, depois verifica as filas virtuais com pacotes de maior tamanho.

Por fim, é necessário assegurar que cada vizinho a quem o pacote codificado é encaminhado tenha alta probabilidade de decodificação. Para cada pacote na sua fila de saída, o nó de repasse estima a probabilidade de seus vizinhos já terem escutado tal pacote. As vezes o nó pode ter certeza, através de informes de recepção, mas se isso não acontecer, o nó faz um levantamento do cálculo da probabilidade de entrega através do protocolo de roteamento. O nó usa essa estimativa

para assegurar que o pacote codificado é decodificável por todos os seus próximos saltos com alta probabilidade. Suponha que um dado nó codifica n pacotes juntos. A probabilidade de que um próximo salto tenha escutado o pacote i é P_i . Então a probabilidade P_D de decodificação do pacote nativo por esse vizinho será dada pela probabilidade dele ter ouvido os demais $n-1$ pacotes nativos combinados, ou seja,

$$P_D = \prod_{i=1}^{n-1} P_i.$$

Considere isso como um passo intermediário enquanto busca por candidatos a codificação. Após reunir os candidatos a codificação COPE verifica se a probabilidade de decodificação P_D , para cada próximo salto, é maior que um limiar de 0,8. Em [5], o valor 0,8 foi considerado como limiar ótimo. Cada próximo salto pode decodificar seu pacote com a probabilidade mínima igual ao limiar.

2.4.5 Decodificação de Pacotes

Para a decodificação de pacotes, em cada nó há um reservatório de pacotes no qual ele mantém uma cópia dos pacotes nativos que recebeu ou enviou. Os pacotes são armazenados em uma tabela *hash* indexados por seus ids. Quando um nó recebe um pacote codificado constituído de n pacotes nativos, ele verifica os ids dos pacotes e os retira do reservatório, após obter os $n-1$ pacotes, o nó decodifica o pacote codificado e obtém o pacote nativo destinado a si.

2.4.6 Formato de pacote

Se o protocolo de roteamento tem seu próprio cabeçalho, COPE insere seu cabeçalho entre os cabeçalhos de roteamento e da MAC. Caso contrário ele se situa entre o cabeçalho da MAC e da camada de rede.

A Fig.2.5 mostra os campos do cabeçalho de COPE [5]. O cabeçalho é dividido em 3 blocos: **Ids dos pacotes nativos codificados**(PKT_ID): O primeiro bloco registra os dados para permitir decodificação de pacotes. Ele começa com a variável ENCODED_NUM, que é o número de pacotes nativos codificados. Para cada pacote nativo o cabeçalho lista seus ids, que é um *hash* de 32 bits que é o endereço IP da fonte e o número de sequência do IP. Este é seguido pelo endereço MAC do próximo salto do pacote. Quando um nó escuta um pacote codificado, ele verifica a lista de próximos saltos para determinar se ele é um destino desejado por algum dos pacotes nativos que compõem o pacote combinado, e nesse caso o pacote é decodificado e processado.

Informes de Recepção: Os informes de recepção constituem o segundo bloco do cabeçalho, como mostra a Fig.2.5. O bloco inicia com o número de informes no cabeçalho REPORT_NUM, cada informe especifica a fonte do pacote relatado SCR_IP. Este é seguido pelo número de sequência IP do último pacote escutado por aquela fonte LAST_PKT, e um mapa de bits dos pacotes escutados recentemente. Por exemplo, um informe é da forma {128.0.1.9, 50, 10000001}, e significa que o último pacote que o nó ouviu da fonte 128.0.1.9 é o pacote 50, também ouviu os pacotes 42

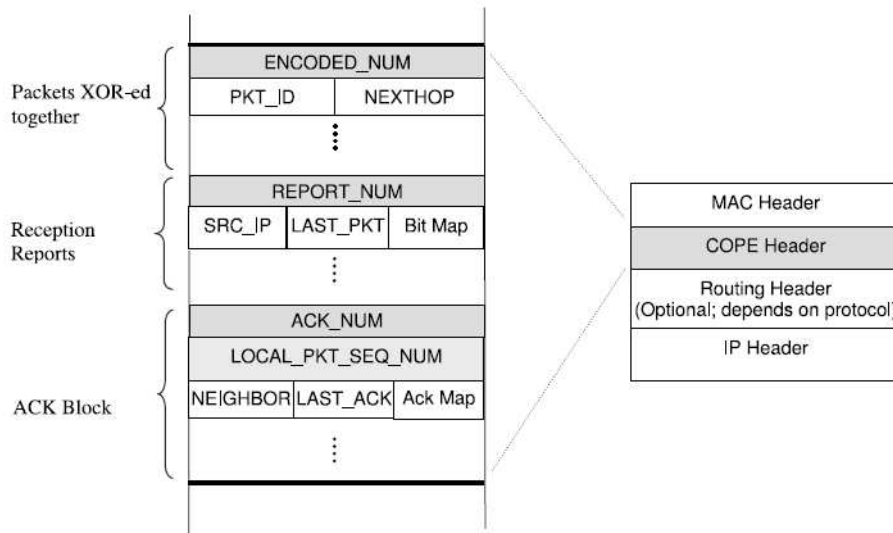


Figura 2.5: Estrutura do cabeçalho COPE. Figura retirada de [5].

e 49, mas nenhum entre eles. O mapa de bits permite ao nó relatar cada pacote múltiplas vezes com a mínima despesa de banda.

Acks Assíncronos: Para assegurar a entrega dos acks com o mínimo de uso de banda, são empregados acks acumulativos. Cada nó mantém um contador de 16-bits por vizinho, chamado `NEIGHBOR_SEQNO_COUNTER`. Sempre que um nó envia um pacote para aquele vizinho, esse contador é incrementado e este valor é atribuído ao pacote como um número de sequência local, `LOCAL_PKT_SEQ_NUM`. Os dois vizinhos usam esse número de sequência para identificar o pacote. Cada pacote codificado contém um cabeçalho ack como mostra a Fig.2.5. O bloco de acks começa com o número de entradas acks, seguido pelo número de sequência local do pacote. Cada entrada ack inicia com o endereço MAC do vizinho. Este é seguido por um ponteiro que indica ao vizinho onde os acks acumulados param, e o mapa de bits indica os pacotes perdidos e recebidos previamente. Por exemplo, uma entrada seria {A, 50, 01111111}, onde os acks para no pacote 50, o mapa de bits mostra que os pacotes de 43-49 foram recebidos e que o 42 foi perdido, sendo todos estes foram enviados pela fonte A.

2.4.7 Controle de fluxo

A Fig.2.6 mostra os procedimentos realizados pelo emissor e receptor para COPE. Sempre que a camada MAC tem a possibilidade de transmitir, o nó retira o pacote do início da fila de saída e procura os candidatos a codificação. Se o nó pode combinar vários pacotes nativos em um pacote codificado, primeiramente são programadas as retransmissões, em seguida adicionados os acks e informes de recepção como mostrado na Fig.2.6(a). Pelo lado do receptor, Fig.2.6(b), quando um pacote chega o nó extrai alguns acks enviados por seus vizinhos. Também extrai alguns informes de recepção e atualiza o estado de seus vizinhos, ou seja, quais pacotes eles armazenam. Em seguida o nó verifica se o pacote o tem como destino. Se o nó não é um próximo salto para o pacote, ele é

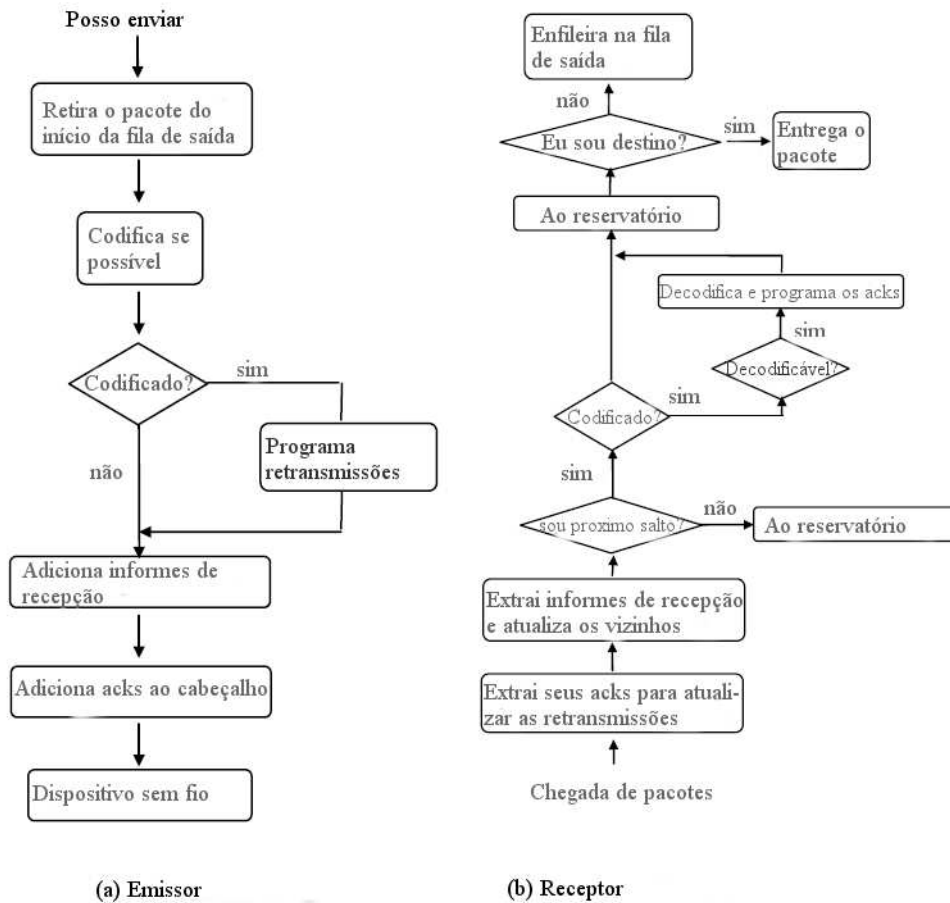


Figura 2.6: Procedimentos na transmissão e recepção em COPE. Figura retirada de [5].

encaminhado para o reservatório de pacotes. Caso contrário, o nó verifica se o pacote é codificado. Se é, o nó tenta decodificar o pacote realizando operações XOR com os pacotes do reservatório que tem seus ids no cabeçalho COPE do pacote codificado. Após decodificar, o nó programa os acks, em seguida armazena o pacote decodificado no reservatório de pacotes. Por fim o nó verifica se é o destino final do pacote, caso seja, o pacote é encaminhado as camadas superiores da pilha de rede, senão, é colocado na fila de saída do nó.

Capítulo 3

Desenvolvimento

Neste capítulo serão apresentados o modelo analítico para a obtenção das métricas de desempenho desejadas e os resultados numéricos.

3.1 Introdução

É certo que a introdução de nós intermediários na rede aumenta a qualidade da comunicação sem fio, podendo-se melhorar a capacidade do sistema substituindo um enlace de baixa qualidade e longa distância por vários nós intermediários com enlaces de alta qualidade e curta distância. Para o trabalho aqui proposto temos apenas um nó intermediário ligando dois grupos de nós usuários. Com o uso da técnica de *Network Coding*, o nó intermediário é capaz de combinar pacotes que foram enviados pelos grupos e repassar pacotes codificados para seus destinos de tal maneira que estes possam decodificá-los. Dessa forma, percebemos primeiramente que reduzimos o número de pacotes enviados mas não reduzimos a informação. Pelo contrário, aumentamos a informação por pacote mantendo o seu tamanho original. Por conseguinte, podemos notar também que, como estamos usando um meio compartilhado, uma única transmissão de um pacote codificado carrega em si a informação de dois pacotes nativos. Dessa forma, tal técnica reduz o atraso de entrega da informação, uma vez que seriam necessárias duas transmissões para cada destino obter seu pacote.

Há diversas formas de realizar codificação de pacotes. No entanto utilizamos aqui algumas características da técnica COPE [5], que, em primeira instância realiza a codificação com uma operação XOR entre os bits de cada pacote nativo gerando o pacote codificado. Em seguida temos um processo chamado de *escuta oportuna*, quando um nó dentro de um grupo transmite um pacote, todos os outros nós desse grupo o escutam e o guardam em um buffer a fim de ajudar na decodificação de um pacote codificado. As principais contribuições deste trabalho são listadas abaixo:

- Um modelo analítico para as métricas de slot vazio e colisão, para o S-ALOHA com e sem *Network Coding*. Comparar as métricas para os dois sistemas através de soluções numéricas.
- Prover descrições de provas embasadas nos lemas e teoremas citados em [1].

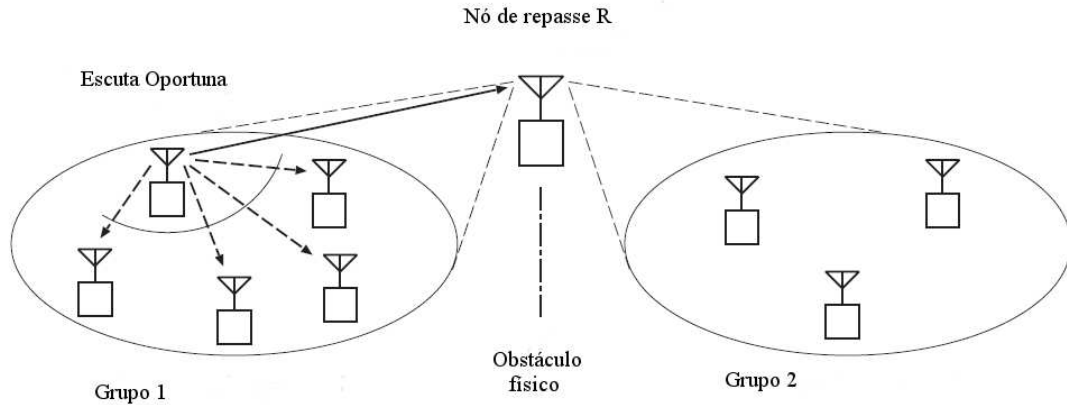


Figura 3.1: Topologia da rede. Figura retirada de [1].

3.2 Descrição do sistema

Esta seção descreve o modelo de rede, o modelo de tráfego, os protocolos S-ALOHA e S-ALOHA/NC e parâmetros para o projeto do sistema.

3.2.1 A topologia da rede

Vamos considerar dois grupos, 1 e 2 de nós usuários estáticos que desejam trocar pacotes de informação através de um nó intermediário R, como na Fig.3.1. Um nó não destina pacotes para outro nó do mesmo grupo. Definimos $v \in \{1,2\}$ e \bar{v} seu complemento. O tráfego é definido como o número médio de novas transmissões e retransmissões dentro de um slot. Definimos o número de usuários em um grupo v como $m_v \geq 1$ para algum $v \in \{1,2\}$. Vamos considerar que todos os nós em um grupo v têm a mesma intensidade de tráfego, que será denotada por $0 < g_v \leq 1$ para algum $v \in \{1,2\}$. Então definimos o tráfego total em um grupo como $G_v = m_v g_v$ para algum $v \in \{1,2\}$. Indicamos também um pacote, gerado no grupo v , como um pacote v para algum $v \in \{1,2\}$.

Cada nó tem um transceptor com uma única antena. Um dado nó ou está transmitindo ou recebendo. Qualquer nó tem uma linha de visada com o nó intermediário R. Porém dois nós usuários de grupos diferentes não sofrem interferência um do outro devido a um obstáculo físico. Qualquer nó usuário em um grupo pode receber um pacote com sucesso enviado pelo nó R desde que os demais nós do mesmo grupo estejam sem transmitir durante o recebimento do pacote, independente de qualquer transmissão dentro do outro grupo. O nó R possui uma capacidade de armazenamento infinita. A fila de transmissão em um nó qualquer tem comportamento FIFO, é

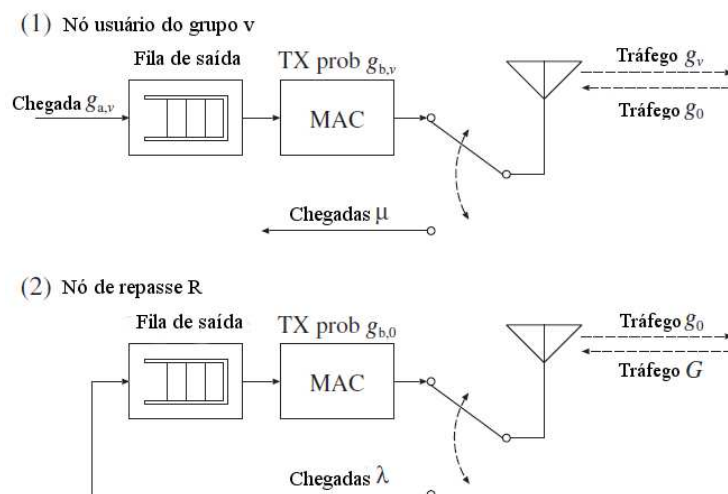


Figura 3.2: Fila de saída e Camada Mac para os protocolos S-ALOHA e S-ALOHA/NC. Figura retirada de [1].

chamada de fila de saída. Se um nó sempre tem pacotes para transmitir na fila de saída, a fila é dita saturada, caso contrário, é dita insaturada.

3.2.2 O protocolo S-ALOHA

O protocolo S-ALOHA é caracterizado por todos os nós estarem sincronizados com o tempo de *slot* e que um pacote de dados só pode ser enviado no início de cada *slot*. Assume-se que tais pacotes têm tamanho constante. A colisão dentro de um grupo ocorre se dois ou mais nós do mesmo grupo transmitem simultaneamente, ou se um nó transmite juntamente com o nó R. Já uma colisão no nó R ocorre se dois ou mais pacotes são transmitidos por ambos os grupos. Considera-se que um pacote só é perdido por colisão, nunca por erros nos bits.

Se um pacote é recebido com sucesso pelo destinatário, um pacote de *acknowledgement* (ACK) é transmitido imediatamente para o transmissor. Considera-se que pacotes ACKs recebidos não possuem erros nos bits, como também os pacotes de dados. O tempo de pacote de dados e o tempo de pacote de ACK são disjuntos dentro de um mesmo *slot*. Quando um pacote é transmitido no começo de um *slot*, o transmissor pode observar se o pacote é recebido com sucesso ou não no fim do *slot* se um ACK é recebido ou não. Se ocorre uma colisão, o pacote é retransmitido até ser recebido com sucesso. Quando um novo pacote chega em um nó e não há pacotes na fila de saída, o nó transmite o pacote imediatamente. Se há uma nova chegada e há pacotes de retransmissão na fila de saída de um nó usuário no grupo v , o nó é dito *backlogged*, e cada pacote é transmitido com uma probabilidade constante $g_{b,v}$ em um *slot*. Perceba que todos os pacotes retransmitidos por um nó usuário são transmitidos com probabilidade $g_{b,v}$ em um slot. O nó R encaminha um pacote com probabilidade $g_{b,0}$ dentro de um slot quando sua fila de saída não está vazia, pois o nó R não gera novos pacotes. O tráfego gerado por um nó usuário em um grupo v é definido como g_v , que será determinado a partir da taxa de chegada $g_{a,v}$ de novos pacotes, da probabilidade de

transmissão $g_{b,v}$, da probabilidade de colisão, e da capacidade da fila. Já o tráfego gerado pelo nó intermediário R é g_0 , e será determinado a partir da taxa de chegada λ de pacotes a serem encaminhados, da probabilidade de transmissão $g_{b,0}$, da probabilidade de colisão, e da capacidade da fila. A Fig.3.2 ilustra como se comportam as filas de saída e camada de acesso ao meio.

3.2.3 O protocolo S-ALOHA/NC

No protocolo S-ALOHA/NC o nó intermediário R combina dois pacotes dos grupos 1 e 2 dentro de um *pacote codificado* e o envia para todos os nós usuários nos grupos 1 e 2. Um pacote não codificado é chamado de *pacote nativo*. Um receptor em um grupo v recebe o pacote codificado e realiza a operação XOR deste com o pacote nativo que foi enviado anteriormente pelo mesmo receptor, dessa forma ele obtém o pacote desejado. É necessário que o nó R saiba se os destinatários podem decodificar os pacotes desejados antes de transmitir pacotes codificados. Os nós usuários devem ouvir os pacotes transmitidos no mesmo grupo e precisam saber qual pacote está combinado dentro do pacote codificado. O nó R pode saber se um usuário tem ou não dado pacote por informes de recepção.

Os nós usuários nos grupos 1 e 2 armazenam os pacotes transmitidos por seus respectivos grupos em uma espécie de reservatório de pacotes, que é uma fila de armazenamento indireto para os pacotes escutados usando escuta oportuna como visto no capítulo anterior. Cada nó mantém uma tabela com os endereços IPs das fontes e número de identificação de IP. Cada nó transmite informes de recepção, os quais incluem informação sobre pacotes armazenados. Essa informação vai dentro do cabeçalho COPE no pacote de dados ou por pacotes de controle. Assim o nó R sabe quais pacotes um nó usuário armazena apenas ouvindo esses informes.

O nó intermediário R mantém duas *filas virtuais* 1 e 2. A fila virtual v armazena os pacotes enviados pelo grupo v . Caso a MAC permita que o nó R transmita, ele verifica se pode gerar um pacote codificado e se os destinos são capazes de decodificá-lo para obter o pacote desejado. Quando um pacote codificado é transmitido, R o envia para ambos os nós destinos. Os IDs dos dois pacotes são combinados e armazenados no cabeçalho COPE dentro do pacote codificado. Se os pacotes são decodificados com sucesso, os nós enviam acks nos sub-slot designado. Dois ACKs habilita o nó R para determinar quais pacotes são apropriados para a próxima transmissão. Caso uma das filas virtuais esteja vazia, e não seja possível gerar um pacote combinado, R transmite um pacote nativo sem esperar uma oportunidade de codificação.

3.3 Análise das métricas de desempenho

Como proposto, não será analisado neste trabalho a vazão, uma vez que este foi feito em [1]. Então serão mapeados os demais eventos a fim de caracterizar os canais de comunicação e encontrar as probabilidades de haver *slot* vazio e colisão no dado sistema, tanto para o protocolo S-ALOHA quanto para o S-ALOHA/NC. Essas medidas permitem um maior entendimento do sistema, sendo

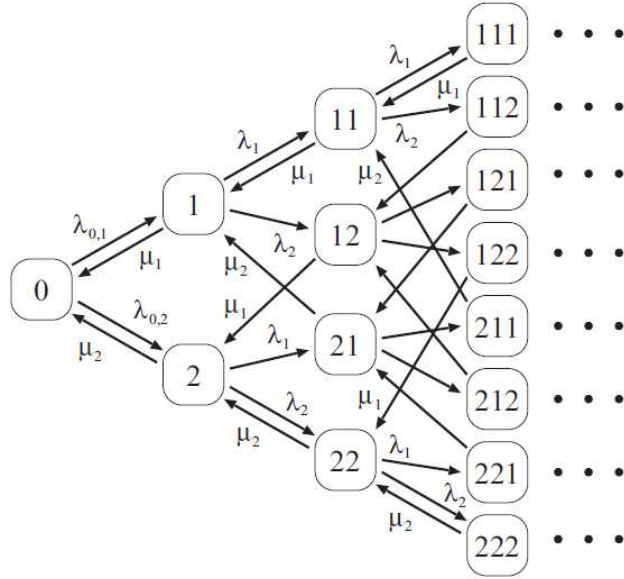


Figura 3.3: Cadeia de Markov para os estados V_k da fila de saída do nó R. Figura retirada de [1].

ferramenta fundamental no aumento da eficiência na rede. Dessa forma manteremos a premissa de que um pacote transmitido por um nó usuário é modelado como uma sequência independente de ensaios de Bernoulli. Então o número de *slots* entre transmissões é geometricamente distribuído com média $1/g_v$ ($v \in \{1,2\}$). Assim constata-se que a probabilidade γ_v de um nó usuário em um grupo v transmitir um pacote com sucesso, dentro de um *slot* é dada pelo produto da probabilidade de apenas um nó qualquer do grupo transmitir e a probabilidade dos demais não transmitirem:

$$\gamma_v = m_v g_v (1 - g_v)^{m_v - 1}. \quad (3.1)$$

A probabilidade de não haver transmissão em um grupo v é expressa como o produto das probabilidades dos nós não transmitirem,

$$\eta_v = (1 - g_v)^{m_v}. \quad (3.2)$$

3.3.1 Protocolo S-ALOHA

Definem-se os estados da fila do nó intermediário R como uma sequência de pacotes $v^n = v_1 v_2 \dots v_n$ ($v_i \in \{1,2\}$) na fila de saída, como na Fig.3.3, e o estado da fila quando está vazia é definido como $v^0 = 0$. O estado da fila no k -ésimo slot é denotado por V_k , para $k \geq 0$, e o estado inicial $V_0 = 0$. Define-se a probabilidade estacionária de um estado V_k como $\varphi(v^n)$, e o número de pacotes v em um estado v^n como $n_v(v^n)$. A probabilidade de transição de chegada, quando a fila está vazia, é dada pelo produto das probabilidades do grupo v enviar um pacote com sucesso e o grupo \bar{v} não transmitir,

$$\lambda_{0,v} = \gamma_v \eta_{\bar{v}}. \quad (3.3)$$

De maneira similar expressam-se as demais probabilidades de transição mostradas na Fig. 3.3,

da seguinte forma. A probabilidade de transição de chegada λ_v , quando a fila não esta vazia, é dada pelo produto da probabilidade do nó R não transmitir, da probabilidade do grupo v transmitir um pacote com sucesso e da probabilidade do grupo \bar{v} não transmitir. Já a probabilidade de transição de partida μ_v , que é a probabilidade de um pacote deixar a fila do nó R é obtida pelo produto da probalidades do nó R transmitir e do grupo \bar{v} não transmitir, logo

$$\lambda_v = (1 - g_{b,0})\gamma_v\eta_{\bar{v}}, \quad (3.4)$$

$$\mu_v = g_{b,0}\eta_{\bar{v}}. \quad (3.5)$$

O fator de utilização com respeito aos pacotes v quando a fila de saída do nó R não está vazia é definido como

$$\rho_v = \frac{\lambda_v}{\mu_v} = \frac{(1 - g_{b,0})\gamma_v}{g_{b,0}}, \quad (3.6)$$

e o fator de utilização total é dado por $\rho = \rho_1 + \rho_2$. Defini-se também o tráfego no nó R, que é a probabilidade de R transmitir dado que tem pacotes, por

$$g_0 = g_{b,0}(1 - \varphi(0)). \quad (3.7)$$

Seja $\varphi(v^*)$ o conjunto de estados em que a fila de saída do nó R tem como primeiro pacote de saída v , podemos expressar a soma das probabilidades estacionárias como

$$\varphi(0) + \varphi(1^*) + \varphi(2^*) = 1, \quad (3.8)$$

e a partir de [1], temos as duas equações que resumem a probabilidade estacionária de todos os estados da cadeia de Markov na Fig.3.3, como demonstrado no Apêndice A no final deste capítulo:

$$\varphi(0) = 1 - \frac{\gamma_1 + \gamma_2}{g_{b,0}(1 + \gamma_1 + \gamma_2)}, \quad (3.9)$$

$$\varphi(v^*) = \frac{\gamma_v}{g_{b,0}(1 + \gamma_1 + \gamma_2)}. \quad (3.10)$$

Vamos definir ξ_v como sendo a probabilidade de não haver transmissão no sistema constituído pelos nós usuários do grupo v e o nó R, ou seja, slot vazio para esse sistema. Dessa forma, usando a lei da probabilidade total temos que,

$$\xi_v = \eta_v((1 - g_{b,0})(1 - \varphi(0)) + (1 - g_{b,0})(\varphi(0))). \quad (3.11)$$

Esta métrica é dada pelo produto da probabilidade do grupo v não transmitir e da probabilidade do nó R também não transmitir. Quando a fila de saída do nó R está vazia, ou seja, a fila encontra-se no estado $\varphi(0)$, a probabilidade desse nó não transmitir é 1. Logo,

$$\xi_v = \eta_v((1 - g_{b,0})(1 - \varphi(0)) + (\varphi(0))), \quad (3.12)$$

e manipulando a equação acima chegamos ao resultado,

$$\xi_v = \eta_v(1 - g_0) = \frac{\eta_v}{1 + \gamma_1 + \gamma_2}. \quad (3.13)$$

Agora iremos mapear os eventos de colisão para o sistema em questão, e definimos σ_v como a probabilidade de colisão que pode ocorrer com os seguintes eventos: dois ou mais nós transmitirem simultaneamente no mesmo grupo, se pelo menos um nó transmitir em cada grupo transmitir no mesmo slot, ou o nó R transmitir e um nó de v também transmitir no mesmo slot. Dessa forma temos que:

$$\begin{aligned} \sigma_v = & \varphi(0)((1 - \eta_v)(1 - \eta_{\bar{v}}) + (1 - \eta_v - \gamma_v)\eta_{\bar{v}}) + \\ & (1 - \varphi(0))((1 - \eta_v)(1 - \eta_{\bar{v}}) + (1 - \eta_v - \gamma_v)\eta_{\bar{v}}(1 - g_{b,0}) \\ & + (1 - \eta_v)\eta_{\bar{v}}g_{b,0}). \end{aligned} \quad (3.14)$$

Nesse ponto é importante entender todos os eventos que mapeiam a métrica de colisão, ressaltando que a análise é feita para o canal de comunicação entre o grupo v e o nó R. O mapeamento foi dividido em duas etapas: a primeira foi analisar as colisões quando a fila do nó R está vazia, sendo representada pela probabilidade estacionária $\varphi(0)$. Nesse caso o termo $(1 - \eta_v)(1 - \eta_{\bar{v}})$ diz respeito a todos os eventos em que há pelo menos uma transmissão em cada grupo simultaneamente, é importante perceber que as transmissões do grupo \bar{v} interferem no desempenho do canal de comunicação entre o grupo v e o nó R, se o grupo v transmitir um pacote e o grupo \bar{v} transmitir um pacote também ocorrerá colisão em R. O próximo termo $(1 - \eta_v - \gamma_v)\eta_{\bar{v}}$ mapeia a colisão dentro do grupo v quando este transmite dois ou mais pacotes simultaneamente e o grupo \bar{v} não transmite. A segunda etapa foi analisar as colisões quando a fila do nó R não está vazia, representada pela probabilidade estacionária $(1 - \varphi(0))$. O termo $(1 - \eta_v)(1 - \eta_{\bar{v}})$ faz o mapeamento dos mesmos eventos da primeira etapa, só que agora o nó R pode ou não transmitir, sendo assim teríamos que a probabilidade dos grupos transmitirem pelo menos um pacote e o nó R não transmitir dado que tem pacotes para transmitir, é dado por $(1 - \eta_v)(1 - \eta_{\bar{v}})(1 - g_{b,0})$, e a probabilidade dos dois grupos transmitirem ao menos um pacote e o nó R transmitir, dado que tem pacotes na fila é $(1 - \eta_v)(1 - \eta_{\bar{v}})g_{b,0}$, somando essas probabilidades temos como resultado $(1 - \eta_v)(1 - \eta_{\bar{v}})$. Isso mostra que quando os dois grupos transmitem pacotes no mesmo *slot*, independente de R transmitir ou não o mapeamento é $(1 - \eta_v)(1 - \eta_{\bar{v}})$. O termo seguinte $(1 - \eta_v - \gamma_v)\eta_{\bar{v}}(1 - g_{b,0})$ diz respeito aos eventos em que o grupo v transmite dois ou mais pacotes, o grupo \bar{v} não transmite e R também não transmite, nesse caso estão mapeados as colisões dentro do grupo v , e por fim temos o termo $(1 - \eta_v)\eta_{\bar{v}}g_{b,0}$ onde estão mapeados os eventos em que o grupo v transmite pelo menos um pacote, o grupo \bar{v} não transmite e R transmite. A expressão 3.14 pode ser simplificada para

$$\sigma_v = (1 - \eta_v)(1 - \eta_{\bar{v}}) + \frac{(1 - \eta_v - \gamma_v)\eta_{\bar{v}}}{1 + \gamma_1 + \gamma_2} + \frac{((1 - \eta_v)\eta_{\bar{v}}(\gamma_1 + \gamma_2))}{1 + \gamma_1 + \gamma_2}. \quad (3.15)$$

Após o mapeamento da colisão, ainda há um evento que não foi mapeado, que é dado pela probabilidade do nó R transmitir e o grupo v não transmitir, ou seja, está relacionado ao canal de descida do sistema, ou melhor, está relacionado com a probabilidade do grupo receber um pacote transmitido por R. Este é aqui designado como ϱ_v e expresso por

$$\varrho_v = \eta_v g_0 = \frac{\eta_v(\gamma_1 + \gamma_2)}{1 + \gamma_1 + \gamma_2}. \quad (3.16)$$

Como mostrado em [1], a vazão por grupo é dado por

$$S_v = \gamma_v \eta_{\bar{v}}(1 - g_0) = \frac{\gamma_v \eta_{\bar{v}}}{1 + \gamma_1 + \gamma_2}, \quad (3.17)$$

então se somarmos as equações (3.13), (3.15), (3.16) e (3.17) obtemos

$$\xi_v + \sigma_v + \varrho_v + S_v = 1. \quad (3.18)$$

Consequentemente podemos expressar a colisão da seguinte forma

$$\begin{aligned} \sigma_v &= 1 - \xi_v - \varrho_v - S_v \\ \sigma_v &= 1 - \eta_v - \frac{\gamma_v \eta_{\bar{v}}}{1 + \gamma_1 + \gamma_2}. \end{aligned} \quad (3.19)$$

3.3.2 Protocolo S-ALOHA/NC

Para essa nova abordagem definem-se os estados da fila do nó R como um par ordenado (n_1, n_2) de números de pacotes das filas virtuais 1 e 2. Um estado na fila no k -ésimo slot é definido como $W_k = (W_{1,k}, W_{2,k})$ para algum $k \geq 0$ e assumimos o estado inicial $W_0 = (0, 0)$. A Fig.(3.4) mostra a cadeia de markov com os estados W_k , onde as transições para o próprio estado não estão esboçadas. A probabilidade estacionária de um estado (n_1, n_2) é denotado como $\psi(n_1, n_2)$ e o estado n_v da fila virtual v é denotado por $\psi(n_v)$. As probabilidades de transição $\lambda_{0,v}$, λ_v e μ_v na figura são expressas como no Protocolo S-ALOHA, nas equações (3)-(5), respectivamente. A transição de um estado $W_k = (n_1, n_2)$ para um estado $W_{k+1} = (n_1 - 1, n_2 - 1)$ significa que o pacote codificado foi entregue com sucesso para ambos os grupos. Tal probabilidade de transição é definida como μ , que é o produto das probabilidades do nó R transmitir e os dois grupos não transmitirem, expressa por

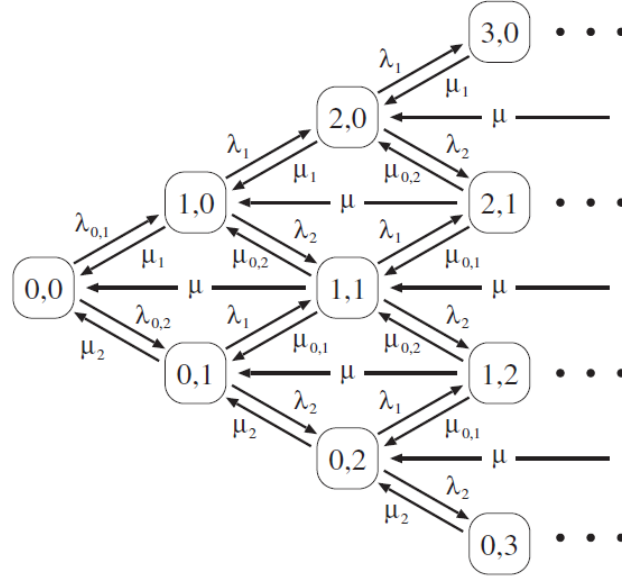


Figura 3.4: Cadeia de Markov para os estados W_k da fila de saída do nó R. Figura retirada de [1].

$$\mu = g_{b,0} \eta_1 \eta_2. \quad (3.20)$$

A probabilidade de transição de um k -ésimo estado $W_{v,k} = n_v > 0$ e $W_{\bar{v},k} = n_{\bar{v}} > 0$ para um estado $k+1$ $W_{v,k+1} = n_v - 1$ e $W_{\bar{v},k+1} = n_{\bar{v}}$ na figura, significa que o pacote v dentro do pacote codificado foi recebido com sucesso, porém \bar{v} não. Tal resultado é obtido pelo produto das probabilidades do nó R transmitir, do grupo v realizar ao menos uma transmissão e o grupo \bar{v} não transmitir. Dessa forma temos

$$\mu_{0,v} = g_{b,0}(1 - \eta_v)\eta_{\bar{v}}, \quad (3.21)$$

e o fator de utilização ρ_v é definido igualmente como na Eq. (3.6) para o protocolo S-ALOHA. O tráfego do nó R é dado por

$$g_0 = g_{b,0}(1 - \psi(0, 0)). \quad (3.22)$$

Para o cálculo da probabilidade de slot vazio definimos ξ_v tal como descrito no S-ALOHA, e temos:

$$\xi_v = \eta_v(1 - g_0), \quad (3.23)$$

Diferentemente do sistema para S-ALOHA, aqui a probabilidade de transmissão $g_{b,0}$ é um parâmetro do sistema. A probabilidade estacionária do estado inicial $W_0 = (0, 0)$, demonstrada no Apêndice B no final do capítulo, é dada por

$$\psi(0, 0) = \frac{(g_{b,0} - (1 - g_{b,0})\gamma_1)(g_{b,0} - (1 - g_{b,0})\gamma_2)}{g_{b,0}(g_{b,0}(1 + \gamma_1 + \gamma_2) - (1 - g_{b,0})\gamma_1\gamma_2)}. \quad (3.24)$$

Assim temos que a probabilidade de slot vazio é dada por

$$\xi_v = \frac{\eta_v g_{b,0}}{g_{b,0}(1 + \gamma_1 + \gamma_2) - (1 - g_{b,0})\gamma_1\gamma_2}. \quad (3.25)$$

Dessa forma podemos verificar o caso particular quando a probabilidade de transmissão $g_{b,0}$ se aproxima de 1 pela esquerda (1-), a probabilidade de slot vazio se aproxima de

$$\lim_{g_{b,0} \rightarrow 1^-} \xi_v = \frac{\eta_v}{1 + \gamma_1 + \gamma_2}, \quad (3.26)$$

que é a probabilidade de slot vazio com o protocolo S-ALOHA. Para explicar melhor tal probabilidade, define-se uma função $\Gamma_v = \frac{\gamma_v}{1 + \gamma_v}$ para $v \in \{1, 2\}$. A partir do Lema 2 e Eq.(54) de [1], podemos fazer a seguinte aproximação

$$\lim_{g_{b,0} \rightarrow \Gamma_{\theta+}} \xi_v = \lim_{g_{b,0} \rightarrow \Gamma_{\theta-}} \xi_v = \frac{\eta_v}{1 + \gamma_{\theta}}, \quad (3.27)$$

onde θ é um indicador também definido em [1] do seguinte modo

$$\theta = 1 \quad \text{se} \quad \gamma_1 \geq \gamma_2$$

ou

$$\theta = 2 \quad \text{se} \quad \gamma_1 < \gamma_2.$$

Mapeando os eventos de colisão para um sistema formado pelo nó R e o grupo v tal qual para o sistema S-ALOHA, temos

$$\sigma_v = (1 - \eta_v)(1 - \eta_{\bar{v}}) + (1 - \eta_v - \gamma_v)\eta_{\bar{v}}(1 - g_0) +$$

$$(1 - \eta_v)\eta_{\bar{v}}g_0. \quad (3.28)$$

Quando a probabilidade $g_{b,0}$ se aproxima de 1- a probabilidade de colisão se aproxima de

$$\lim_{g_{b,0} \rightarrow 1^-} \sigma_v = (1 - \eta_v)(1 - \eta_{\bar{v}}) + \frac{(1 - \eta_v - \gamma_v)\eta_{\bar{v}}}{1 + \gamma_1 + \gamma_2} + \frac{((1 - \eta_v)\eta_{\bar{v}}(\gamma_1 + \gamma_2))}{1 + \gamma_1 + \gamma_2}, \quad (3.29)$$

que é a probabilidade de colisão do sistema com protocolo S-ALOHA. Agora se $g_{b,0}$ se aproxima de Γ_θ , temos:

$$\lim_{g_{b,0} \rightarrow \Gamma_{\theta+}} \sigma_v = (1 - \eta_v)(1 - \eta_{\bar{v}}) + \frac{(1 - \eta_v - \gamma_v)\eta_{\bar{v}}}{1 + \gamma_\theta} + \frac{(1 - \eta_v)\eta_{\bar{v}}(\gamma_\theta)}{1 + \gamma_\theta}. \quad (3.30)$$

Como no sistema S-ALOHA, ainda há um evento a ser mapeado, que é a probabilidade de R transmitir e o grupo v não transmitir, análogo ao mencionado na Eq. 3.16, designado também por ϱ_v e dado por

$$\lim_{g_{b,0} \rightarrow \Gamma_\theta} \varrho_v = \eta_v g_{b,0}(1 - \varphi(0,0)) = \eta_v g_0 = \frac{\eta_v \gamma_\theta}{1 + \gamma_\theta}. \quad (3.31)$$

Como as demais métricas, esta também se caracteriza por se aproximar da probabilidade de colisão do sistema com protocolo S-ALOHA quando

$$\lim_{g_{b,0} \rightarrow 1^-} \varrho_v = \frac{\eta_v(\gamma_1 + \gamma_2)}{1 + \gamma_1 + \gamma_2}. \quad (3.32)$$

Análogo a Eq.(3.17) a vazão é dada por

$$S_v = \gamma_v \eta_{\bar{v}}(1 - g_0) = \frac{\gamma_v \eta_{\bar{v}}}{1 + \gamma_\theta} \quad (3.33)$$

Sabendo que $S_v + \varrho_v + \xi_v + \sigma_v = 1$, das equações (3.27), (3.31) e (3.33), podemos expressar a probabilidade de colisão da forma:

$$\sigma_v = 1 - \eta_v - \frac{\gamma_v \eta_{\bar{v}}}{1 + \gamma_\theta} \quad (3.34)$$

3.4 Apêndices

3.4.1 Apêndice A - Demonstração das Equações (3.9) e (3.10)

Podemos representar todas as probabilidades estacionárias dos estados da cadeia de Markov da Fig.3.3 por $\varphi(0)$, $\varphi(1^*)$ e $\varphi(2^*)$. A probabilidade estacionária $\varphi(1^*)$ representa a soma de todas as probabilidades estacionárias dos estados em que o primeiro pacote da fila do nó R é um pacote do grupo 1, e $\varphi(2^*)$ segue explicação análoga, dessa forma temos que

$$\varphi(0) + \varphi(1^*) + \varphi(2^*) = 1. \quad (3.35)$$

A razão entre as probabilidades estacionárias de $\varphi(1^*)$ e $\varphi(2^*)$ é proporcional à $\lambda_{0,1}/\lambda_{0,2} = \lambda_1/\lambda_2$ e inversamente proporcional à μ_1/μ_2 , assim

$$\frac{\varphi(1^*)}{\varphi(2^*)} = \frac{\lambda_1}{\lambda_2} \cdot \frac{\mu_2}{\mu_1} = \frac{\gamma_1 \eta_2}{\gamma_2 \eta_1} \cdot \frac{\eta_1}{\eta_2} = \frac{\gamma_1}{\gamma_2}. \quad (3.36)$$

Substituindo (3.36) em (3.35) obtemos

$$\varphi(v^*) = \frac{\lambda_v}{\lambda_1 + \lambda_2}(1 - \varphi(0)), \quad (3.37)$$

para $v \in \{1, 2\}$. Sabendo que o nó R não gera pacotes e assumindo que a probabilidade estacionária $\varphi(0)$ é positiva, logo a fila de saída é insaturada, a taxa de chegada λ_R e a taxa de partida μ_R do nó de repasse R são iguais. Temos que a taxa de chegada λ_R é dada por

$$\lambda_R = \varphi(0)(\lambda_{0,1} + \lambda_{0,2}) + (1 - \varphi(0))(\lambda_1 + \lambda_2). \quad (3.38)$$

Substituindo (3.3) e (3.4) em (3.38) concluímos que

$$\lambda_R = (\gamma_1\eta_2 + \gamma_2\eta_1)(1 - g_{b,0}(1 - \varphi(0))) = (\gamma_1\eta_2 + \gamma_2\eta_1)(1 - g_0). \quad (3.39)$$

A taxa de partida do nó R μ_R é dada por

$$\mu_R = \mu_1\varphi(1^*) + \mu_2\varphi(2^*), \quad (3.40)$$

substituindo (3.5) e (3.36) em (3.40) obtemos

$$\mu_R = \frac{g_{b,0}(1 - \varphi(0))(\gamma_1\eta_2 + \gamma_2\eta_1)}{\gamma_1 + \gamma_2}. \quad (3.41)$$

Igualando as equações (3.39) e (3.41) temos que

$$\varphi(0) = 1 - \frac{\gamma_1 + \gamma_2}{g_{b,0}(1 + \gamma_1 + \gamma_2)}, \quad (3.42)$$

e substituindo (3.42) em (3.36) concluímos

$$\varphi(v^*) = \frac{\gamma_v}{g_{b,0}(1 + \gamma_1 + \gamma_2)}. \quad (3.43)$$

3.4.2 Apêndice B - Demonstração da Equação (3.24)

A partir da Fig. 3.4, utilizando equações de balanceamento obtemos

$$\psi(1, 0) = \frac{\rho_1}{1 - g_{b,0}}\psi(0, 0), \quad \psi(0, 1) = \frac{\rho_2}{1 - g_{b,0}}\psi(0, 0). \quad (3.44)$$

Estendendo tal relação para os demais estados da cadeia de Markov temos

$$\psi(n_1 + 1, n_2) = \rho_1\psi(n_1, n_2), \quad \psi(n_1, n_2 + 1) = \rho_2\psi(n_1, n_2), \quad (3.45)$$

para algum $(n_1, n_2) \neq (0, 0)$. Das equações (3.44) e (3.45) obtemos a probabilidade estacionária de um estado qualquer em função de $\psi(0, 0)$

$$\psi(0, 0) = \frac{\rho_1^{n_1}\rho_2^{n_2}\psi(0, 0)}{1 - g_{b,0}}, \quad (3.46)$$

para algum $(n_1, n_2) \neq (0, 0)$. Somando todas as probabilidades estacionárias, as quais estão em função de $\psi(0, 0)$, e usando a condição de normalização

$$1 = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \psi(n_1, n_2) = \frac{\psi(0, 0)}{1 - g_{b,0}} \left(\sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} \rho_1^{n_1}\rho_2^{n_2} - g_{b,0} \right), \quad (3.47)$$

manipulando a equação temos

$$1 = \frac{\psi(0,0)(1 - g_{b,0}(1 - \rho_1)(1 - \rho_2))}{(1 - g_{b,0})(1 - \rho_1)(1 - \rho_2)}. \quad (3.48)$$

Dessa forma podemos obter a probabilidade estacionária $\psi(0,0)$, expressa como

$$\psi(0,0) = \frac{(g_{b,0} - (1 - g_{b,0})\gamma_1)(g_{b,0} - (1 - g_{b,0})\gamma_2)}{g_{b,0}(g_{b,0}(1 + \gamma_1 + \gamma_2) - (1 - g_{b,0})\gamma_1\gamma_2)}. \quad (3.49)$$

Capítulo 4

Resultados Numéricos

Este capítulo apresenta os resultados obtidos durante a simulação do trabalho e seus pontos mais importantes.

4.1 Introdução

Para obter as curvas de probabilidades vamos manter a mesma configuração de [1], considerando uma rede com número de usuários no grupo 1, $m_1 = 5$ e o número de usuários no grupo 2, $m_2 = 1$. Visando manter um fluxo de dados justo entre os grupos e o nó R, a Eq(27) de [1] expressa uma relação entre os tráfegos dos nós, obtida por igualar S_1 a S_2 , dada por $g_2 = \frac{m_1 g_1}{m_1 g_1 + m_2 (1 - g_1)}$. Assim temos os parâmetros necessários para esboçar os gráficos e compararmos os dois sistemas. É importante frisar que, embora a probabilidade de transmissão do nó de repasse R não seja parâmetro para encontrar as métricas de desempenho no sistema S-ALOHA, ela precisa ter um valor bem estabelecido, até porque a estabilidade da fila de saída desse nó depende da vazão de pacotes dada por ele. Dessa forma, temos que o fator de utilização desse sistema $\rho = \rho_1 + \rho_2 < 1$ para manter a estabilidade da fila de R. Com isso temos que

$$\rho = \rho_1 + \rho_2 < 1. \quad (4.1)$$

Da definição de fator de utilização em 3.6 obtemos

$$\frac{\lambda_1}{\mu_1} + \frac{\lambda_2}{\mu_2} < 1, \quad (4.2)$$

agora substituindo as probabilidades de transições de chegada e partida como nas equações (3.4) e (3.5) temos

$$\frac{(1 - g_{b,0})\gamma_1}{g_{b,0}} + \frac{(1 - g_{b,0})\gamma_2}{g_{b,0}} = \frac{(1 - g_{b,0})(\gamma_1 + \gamma_2)}{g_{b,0}} < 1, \quad (4.3)$$

e manipulando a equação chegamos a seguinte expressão:

$$g_{b,0} > \frac{\gamma_1 + \gamma_2}{1 + \gamma_1 + \gamma_2}, \quad (4.4)$$

essa fórmula garante que a fila será insaturada.

De maneira similar, temos que para o sistema com protocolo S-ALOHA/NC, é necessário manter a estabilidade da fila. Como visto, neste caso a probabilidade de transmissão do nó de repasse R é um parâmetro do sistema, parâmetro este que pode ser aproximado pela função $\Gamma_\theta = \frac{\gamma_\theta}{1+\gamma_\theta}$. É preciso entender o porque dessa aproximação. No sistema em questão, o nó R mantém duas filas virtuais, uma para cada grupo de nós usuários, e com isso é preciso garantir a estabilidade de ambas as filas. Como a probabilidade de transmissão do nó R $g_{b,0}$ está relacionada ao indicador θ , que tem valor 1 se $\gamma_1 \geq \gamma_2$ e valor 2 caso contrário, a probabilidade de transmissão do nó R irá variar de acordo com a maior probabilidade de transmissão de um pacote com sucesso entre os dois grupos. Assim, se a estabilidade for mantida para a fila virtual do grupo que transmite mais pacotes, com certeza ela será mantida para a outra fila também. Com isso é suficiente analisar a desigualdade $\rho_\theta < 1$ para chegarmos ao seguinte resultado

$$\rho_\theta < 1 \rightarrow \frac{(1 - g_{b,0})\gamma_\theta}{g_{b,0}} < 1. \quad (4.5)$$

Da definição de fator de utilização na Eq. (3.6) temos

$$\rho_\theta = \frac{\lambda_\theta}{\mu_\theta}, \quad (4.6)$$

agora substituindo as probabilidades de transições de chegada e partida analogamente a Eq.(4.3) chegamos à desigualdade

$$\frac{(1 - g_{b,0})\gamma_1}{g_{b,0}} < 1. \quad (4.7)$$

Manipulando a equação obtemos

$$g_{b,0} > \frac{\gamma_\theta}{1 + \gamma_\theta}. \quad (4.8)$$

Essa fórmula garante a insaturação das filas virtuais.

4.2 Análise dos resultados

Por fim podemos comparar as métricas de desempenho a partir das expressões desenvolvidas no Cap.3, a seguir temos as curvas em função do tráfego gerado pelos grupos de usuários e uma visão geral sobre todo o sistema.

A Fig.4.1 mostra as curvas de probabilidade de transmissão para o nó de repasse R em ambos os sistemas, de acordo com as equações (4.4) e (4.8). Percebe-se que a probabilidade de R transmitir no sistema com protocolo S-ALOHA é maior que no sistema com protocolo S-ALOHA/NC, o que é bastante coerente, pois o primeiro não faz combinação de pacotes, logo precisa transmitir bem mais para dar vazão aos pacotes da fila de saída. Essas curvas de probabilidade de transmissão são importantes para entender melhor a comparação entre as métricas de desempenho nas figuras a seguir. As métricas são comparadas por grupo, ou seja, para cada grupo de usuários são analisadas as medidas de desempenho com os dois protocolos propostos, S-ALOHA e S-ALOHA/NC.

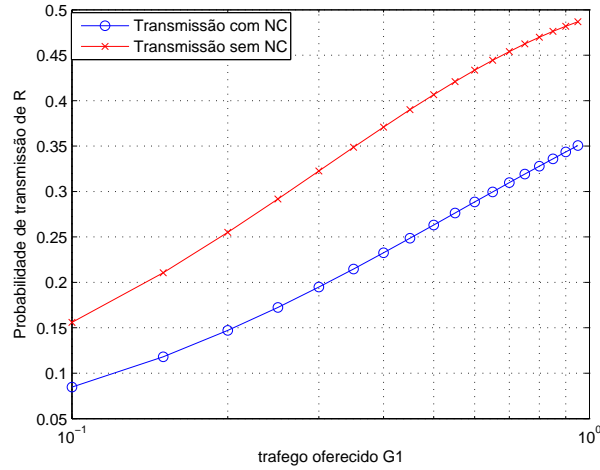


Figura 4.1: Comparação entre as probabilidades de transmissão do nó R.

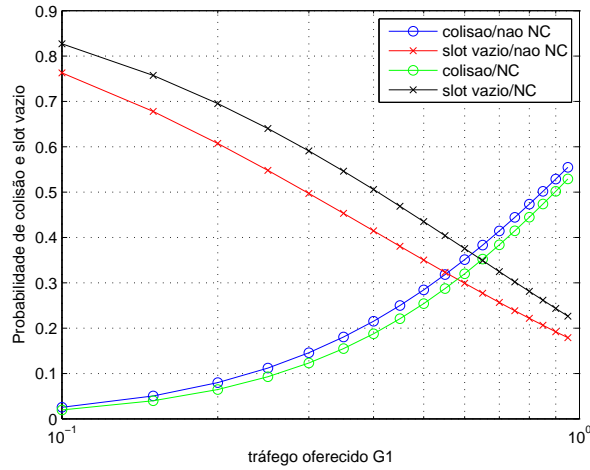


Figura 4.2: Probabilidades de colisão e slot vazio para o grupo 1.

A Fig.4.2 apresenta as curvas de probabilidades de slot vazio e colisão para o subsistema formado pelo conjunto de usuários do grupo 1 e o nó de repasse R para ambos os protocolos. A comparação revela uma queda na taxa de colisão e o aumento da ociosidade da rede à medida que o tráfego na rede diminui.

A Fig.4.3 revela uma queda na probabilidade do evento relacionado ao downlink desse subsistema. Este evento está relacionado aos pacotes transmitidos por R. Analisando o canal entre o nó R e o grupo 1, quando um pacote do grupo 2 é repassado por R para o grupo 1 temos que o canal está sendo usado com eficiência, pois um novo pacote é recebido pelo grupo. Contudo se R transmite um pacote do grupo 1, o canal de comunicação entre R e o grupo 1 é utilizado de maneira ineficiente, pois envia um pacote que o grupo já tem, porém nesse mesmo slot de transmissão o canal de comunicação entre R e o grupo 2 recebe um pacote do grupo 1 e este canal é bem utilizado. A técnica de codificação em rede utiliza esse evento para aumentar a vazão na rede, combinando

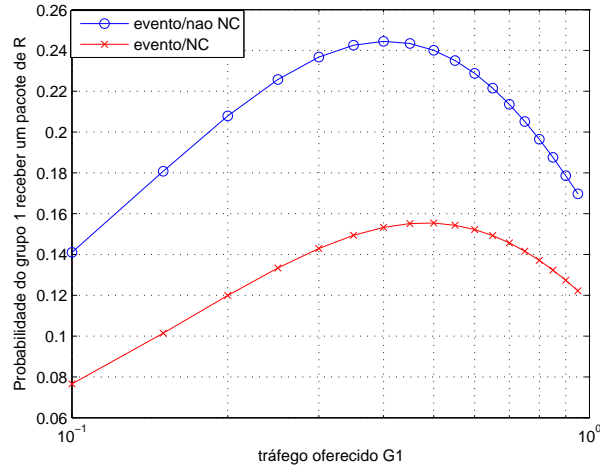


Figura 4.3: Probabilidade do nó R transmitir um pacote no subsistema formado pelo grupo 1 e o nó R.

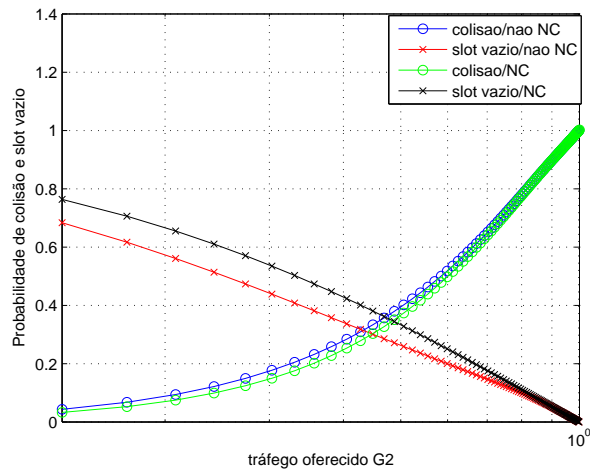


Figura 4.4: Probabilidades de colisão e slot vazio para o grupo 2.

os pacotes no nó R, ambos os grupos podem receber seus pacotes designados com apenas uma transmissão de R, garantindo que o meio compartilhado seja utilizado eficientemente para todos os destinatários.

A Fig.4.4 faz o levantamento das curvas de probabilidades de slot vazio e colisão para o subsistema formado pelo conjunto de usuários do grupo 2 e o nó de repasse R também para ambos os protocolos. Ao compararmos S-Aloha com S-Aloha/NC, tal qual a Fig.4.2, verificamos um diminuição na taxa de colisão e aumento da ociosidade da rede.

A Fig.4.5 também revela uma queda na probabilidade do evento relacionado ao downlink nesse canal de comunicação, e tal evento está relacionado à entrega de pacotes do grupo 1 através do nó R. Por fim, a Fig.4.6 esboça as curvas para vazão nos dois subsistemas com ambos os

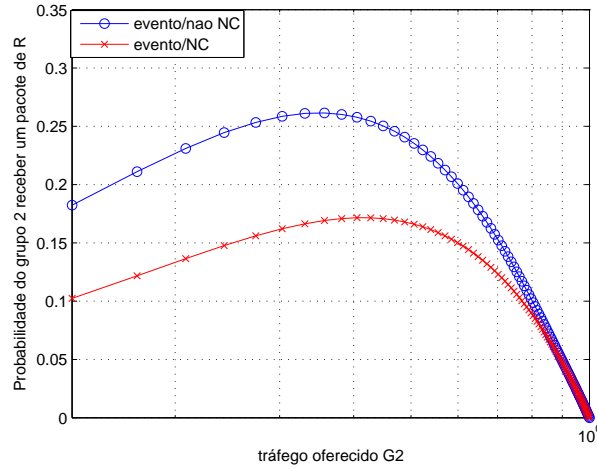


Figura 4.5: Probabilidade do nó R transmitir um pacote no subsistema formado pelo grupo 2 e o nó R.

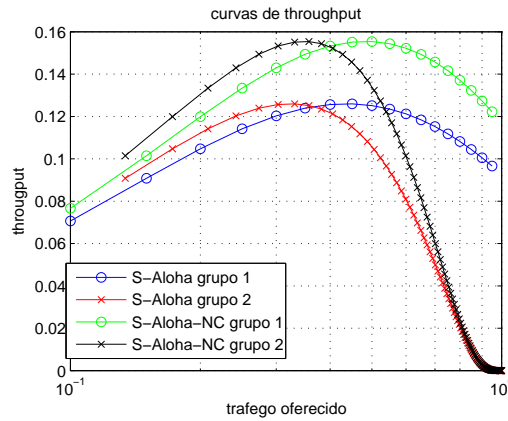


Figura 4.6: Curvas de vazão.

protocolos. Como mencionado anteriormente, as curvas são iguais, para tornar mais justo o número de transmissões em cada grupo. As curvas de probabilidade das vazões nos grupos 1 e 2, quando o sistema não emprega *Network Coding*, tem máximos em 0.126 em intensidades de tráfego diferentes. Quando emprega a técnica de codificação, os máximos são obtidos com probabilidade de 0.155, em intensidades distintas de tráfego para cada grupo.

A partir dos dados obtidos através da caracterização dos dois canais de comunicação, o primeiro relacionado ao grupo 1 e o nó R, e o segundo relacionado ao grupo 2 e nó R, os eventos mapeados nos mostram um melhor entendimento da rede como um único sistema. Em primeira instância a probabilidade de transmissão $g_{b,0}$ do nó de repasse R é bem menor quando tal nó está empregando *Network Coding*. Dessa forma, mantendo o mesmo tráfego nos nós usuários, é bastante coerente que a ociosidade da rede aumente pois o tráfego g_0 de R é diretamente proporcional a probabilidade de transmissão. Quanto à colisão, esta apresenta uma diminuição devido ao fato de que no sistema com protocolo S-ALOHA/NC ocorrem menos transmissões. Por fim, o evento relacionado ao

repassa de pacotes pelo nó R nos explica melhor o ganho do sistema. Como pode o nó R transmitir menos e ter maior ganho? É simples, no sistema com protocolo S-ALOHA o nó de repasse só transmite um pacote por slot de tempo, assim quando surge uma oportunidade de transmissão ele verifica na sua fila de saída e retira o primeiro pacote, que pode ser tanto um pacote para o grupo 1 ou 2, e o encaminha. O pacote só interessará a um dos grupos, enquanto que para o outro só ocorrerá ocupação do canal sem informação útil. Com isso a probabilidade de transmissão de R tem que ser alta para dar vazão aos pacotes de ambos os grupos na sua fila de saída, pois a cada transmissão com sucesso efetivada, só um grupo é beneficiado. No caso em que o nó de repasse opera com *Network Coding*, uma transmissão com sucesso de um pacote combinado beneficia ambos os grupos, e mesmo que haja colisão em um dos grupos o outro é beneficiado, dessa forma o nó R não precisa ter uma probabilidade de transmissão muito alta, pois com apenas uma transmissão ele dá vazão a dois pacotes de sua fila de saída. A grande questão é que na maioria das transmissões realizadas por R, quando emprega *Network Coding*, serão enviados pacotes codificados, enquanto que com o protocolo S-ALOHA tradicional ele tende a ficar alternando entre pacotes do grupo 1 e 2, por isso no segundo caso, mesmo transmitindo menos alcança-se maior vazão.

Capítulo 5

Conclusão

Neste trabalho foi desenvolvido um modelo analítico para obter as métricas de desempenho de slot vazio e colisão, dando continuidade ao trabalho proposto por Umehara *et al.* em [1] para um sistema com protocolo S-ALOHA/NC com múltiplos saltos. Seguindo a mesma linha, foram demonstradas as fórmulas necessárias para o levantamento das medidas, baseadas nos lemas e teoremas apresentados por Umehara *et al.*, para ambos os sistemas em consideração, S-ALOHA e S-ALOHA/NC. A comparação consiste em avaliar as métricas e entender o comportamento do sistema como um todo, o que mostrou a eficiência do uso da técnica de codificação *Network Coding* apesar de aumentar a ociosidade da rede, porém verifica-se uma diminuição na taxa de colisão e como já foi visto aumenta a vazão do sistema com a diminuição do tráfego oferecido à rede.

Trabalhos futuros poderão avaliar tais medidas de desempenho para uma topologia em que se inclua mais um grupo de nós usuários ou mais um nó de repasse. Outra linha que pode ser seguida é a análise das métricas para uma configuração ad hoc com o protocolo S-ALOHA/NC com um salto, nesse caso todos os nós empregam *Network Coding* quando houver possibilidade de combinar pacotes.

A principal contribuição deste trabalho é a descoberta da nova métrica de desempenho relacionada ao canal de downlink. Como verificado, o evento que caracteriza tal medida tem menor probabilidade de ocorrer quando empregamos *Network Coding*, contudo a codificação garante maior eficiência do canal, e a cada vez que o evento ocorre há alta probabilidade do canal ser ocupado com informação útil à ambos os grupos. Um estudo interessante seria analisar como se comporta essa métrica à medida que incluirmos mais grupos de usuários.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] UMEHARA, D.; DENNO, S.; MORIKURA, M. Performance analysis of slotted aloha and network coding for single-relay multi-user wireless networks. *Elsevier B.V*, v. 9, p. 164–179, 2010.
- [2] MURTHY, C. *Ad Hoc Wireless Networks Architectures and Protocols*. [S.l.]: Prentice Hall, 2004.
- [3] ABRAMSON, N. The aloha system: Another alternative for computer communications. *Proceedings of the fall joint computer conference AFIPS*, 1970.
- [4] HO, T. *Network Coding: An Introduction*. [S.l.]: Cambridge University Press, 2008.
- [5] KATTI, S.; RAHUL, H.; HU, W. Xor in the air: practical wireless network coding. *IEEE/ACM Transactions on Networking*, v. 16, p. 497–510, 2008.
- [6] RUDOLF, A.; CAI, N.; ROBERT, S.-Y. Network information flow. *IEEE Transactions on Information Theory*, v. 46, p. 1204–1216, 2000.
- [7] YEUNG, R. *Network Coding Theory*. [S.l.]: Now Publishers Inc., 2006.
- [8] DIESTEL, R. *Graph Theory*. [S.l.]: Springer-Verlag Heidelberg, 2005.
- [9] SANDERS, P.; CHOU, A.; EFFROS, M. Polynomial time algorithms for multicast network code construction. *IEEE Transactions on Information Theory*, v. 51, p. 1973–1982, 2006.

ANEXOS

I. PROGRAMAS DE SIMULAÇÃO EM C

I.1 Probabilidades de transmissão do nó de repasse R

I.1.1 Probabilidade de Transmissão $g_{b,0}$ para o caso S-Aloha

```
%Bibliotecas
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

%Função principal

main(){

%declaração de variáveis
float n1,n2,y1,y2,g1,g2,Pr,y;
int m1,m2;

%variável tipo arquivo
FILE *pq;

%inicialização de variáveis
m1=5;
m2=1;
g1=0.02;

if((pq=fopen("probabilidadeTx.txt","a")) == NULL){
printf("\n O arquivo nao pode ser aberto.");
system("pause");
}

while(g1<=0.2){

    g2=(m1*g1)/(m1*g1 + m2*(1-g1));
    %probabilidade de um grupo transmitir com sucesso
    y1=(m1*g1)*pow((1-g1),(m1-1));
    y2=g2;

    %Probabilidade de um grupo não transmitir
    n1=pow((1-g1),m1);
    n2=1-g2;
```

```

    %Probabilidade de transmissão
    Pr=(y1+y2)/(1+y1+y2);

    fprintf(pq, " %f ",Pr);

    g1=g1+0.01;

}

fclose(pq);

}

```

I.1.2 Probabilidade de Transmissão $g_{b,0}$ para o caso S-Aloha/NC

```

%bibliotecas
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

%Função principal
main(){

%Declaração de variáveis
float n1,n2,y1,y2,g1,g2,Pr,y;
int m1,m2;

%Variável tipo ponteiro para arquivo
FILE *aq,*pq;

%inicialização de Variáveis
m1=5;
m2=1;

g1=0.02;

if((pq=fopen("pr_transmissao.txt","a")) == NULL){
printf("\n O arquivo nao pode ser aberto.");
system("pause");
}

while(g1<=0.2){

    g2=(m1*g1)/(m1*g1 + m2*(1-g1));

```

```

%Probabilidade de um grupo transmitir com sucesso
y1=(m1*g1)*pow((1-g1),(m1-1));
y2=g2;

%Probabilidade do grupo não transmitir
n1=pow((1-g1),m1);
n2=1-g2;

if(y1>=y2)
    y=y1;
else
    y=y2;

%Probabilidade de transmissão
Pr=y/(1+y);

fprintf(pq, " %f ",Pr);
g1=g1+0.01;
}
fclose(pq);
}

```

I.2 Métricas de Desempenho

I.2.1 Probabilidade de Slot Vazio caso S-aloha

```

%Bibliotecas
#include<stdio.h>
#include<stdlib.h>

%Função principal
main(){

%declaração de variáveis
float y1,y2,n1,n2,sv1,sv2,g1,g2,G1;
int m1,m2;

%Variável tipo ponteiro para arquivo
FILE *a,*b,*c,*d;

%Inicialização de variáveis
m1=5;
m2=1;

```

```

g1=0.02;

%Gerando arquivos
if((a=fopen("slotvazioG1.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}

if((b=fopen("trafegoG1.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}

if((c=fopen("slotvazioG2.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}

if((d=fopen("trafegoG2.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}

%Computando os valores
while(g2<=1){
    G1=m1*g1;
    g2=(G1)/(G1 + m2*(1-g1));
    y1=G1*pow((1-g1),(m1-1));
    y2=g2;
    n1=pow((1-g1),m1);
    n2=1-g2;
    sv1=n1/(1+y1+y2);
    sv2=n2/(1+y1+y2);
    if(G1<1){
        fprintf(a, " %f ",sv1);
        fprintf(b, " %f ",G1);
    }
    if(g2>0.1){
        fprintf(c, " %f ",sv2);
        fprintf(d, " %f ",g2);
    }
    g1=g1+0.01;
}
fclose(a);
fclose(b);
fclose(c);
fclose(d);
}

```

I.2.2 Probabilidade de Slot Vazio caso S-aloha/NC

```
%Bibliotecas
#include<stdio.h>
#include<stdlib.h>

%Função principal
main(){

%declaração de variáveis
float y1,y2,n1,n2,sv1,sv2,g1,g2,G1;
int m1,m2;

%Variável tipo ponteiro para arquivo
FILE *a,*b,*c,*d;

%Inicialização de variáveis
m1=5;
m2=1;

g1=0.02;

%Gerando arquivos
if((a=fopen("slotvazioNCG1.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}

if((b=fopen("trafegoG1NC.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}

if((c=fopen("slotvazioG2NC.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}

if((d=fopen("trafegoG2NC.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}

%Computando os valores
while(g2<=1){
    G1=m1*g1;

    g2=(G1)/(G1 + m2*(1-g1));

    y1=G1*pow((1-g1),(m1-1));
    y2=g2;
}
```

```

n1=pow((1-g1),m1);
n2=1-g2;

if(y1>=y2)
    y=y1;
else
    y=y2;

sv1=n1/(1+y);
sv2=n2/(1+y);

if(G1<1){
    fprintf(a, " %f ",sv1);
    fprintf(b, " %f ",G1);
}
if(g2>0.1){
    fprintf(c, " %f ",sv2);
    fprintf(d, " %f ",g2);
}
g1=g1+0.01;
}
fclose(a);
fclose(b);
fclose(c);
fclose(d);
}

```

I.2.3 Probabilidades de colisão caso S-aloha

```

%Bibliotecas
#include<stdio.h>
#include<stdlib.h>

%Função principal
main(){

%Declaração de variáveis
float y1,y2,n1,n2,col1,col2,g1,g2,G1;
int m1,m2;
%Variável tipo ponteiro para arquivo
FILE *a,*b,*c,*d;

%Inicialização de variáveis
m1=5;

```

```

m2=1;
g1=0.02;

%Gerando arquivos
if((a=fopen("colisaoG1.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}
if((b=fopen("colisaotrafegoG1.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}
if((c=fopen("colisaoG2.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}
if((d=fopen("colisaotrafegoG2.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}

%Computando as métricas
while(g2<=1){

    G1=m1*g1;

    g2=(G1)/(G1 + m2*(1-g1));

    y1=G1*pow((1-g1),(m1-1));
    y2=g2;

    n1=pow((1-g1),m1);
    n2=1-g2;

    col1 = 1 - n1 - (y1*n2/(1+y1+y2));
    col2 = 1 - n2 - (y2*n1/(1+y1+y2));

    if(G1<1){
        fprintf(a, " %f ",col1);
        fprintf(b, " %f ",G1);
    }
    if(g2>0.1){
        fprintf(c, " %f ",col2);
        fprintf(d, " %f ",g2);
    }
    g1=g1+0.01;
}

```

```

fclose(a);
fclose(b);
fclose(c);
fclose(d);
}

```

I.2.4 Probabilidades de colisão caso S-aloha/NC

```

%Bibliotecas
#include<stdio.h>
#include<stdlib.h>

%Função principal
main(){

%Declaração de variáveis
float y1,y2,n1,n2,col1,col2,g1,g2,G1;
int m1,m2;
%Variável tipo ponteiro para arquivo
FILE *a,*b,*c,*d;

%Inicialização de variáveis
m1=5;
m2=1;
g1=0.02;

%Gerando arquivos
if((a=fopen("colisaoG1NC.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}
    if((b=fopen("colisaotrafegoG1NC.txt","a")) == NULL){
        printf("\n 0 arquivo nao pode ser aberto.");
        system("pause");
    }
if((c=fopen("colisaoG2NC.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}
    if((d=fopen("colisaotrafegoG2NC.txt","a")) == NULL){
        printf("\n 0 arquivo nao pode ser aberto.");
        system("pause");
    }

%Computando as métricas

```



```

while(g2<=1){

    G1=m1*g1;

    g2=(G1)/(G1 + m2*(1-g1));

    y1=G1*pow((1-g1),(m1-1));
    y2=g2;

    n1=pow((1-g1),m1);
    n2=1-g2;

    if(y1>=y2)
        y=y1;
    else
        y=y2;

    col1 = 1 - n1 - (y1*n2/(1+y));
    col2 = 1 - n2 - (y2*n1/(1+y));

    if(G1<1){
        fprintf(a, " %f ",col1);
        fprintf(b, " %f ",G1);
    }
    if(g2>0.1){
        fprintf(c, " %f ",col2);
        fprintf(d, " %f ",g2);
    }
    g1=g1+0.01;
}
fclose(a);
fclose(b);
fclose(c);
fclose(d);
}

```

I.2.5 Probabilidade do evento de downlink caso S-Aloha

```

%Bibliotecas
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

%Função principal
main(){

```

```

%Declaração de variáveis
float n1,n2,y1,y2,g1,g2,Pr1,Pr2;
int m1,m2;

%Variável tipo ponteiro para arquivo
FILE *aq,*pq;

%Inicialização de variáveis
m1=5;
m2=1;
g1=0.02;

%Gerando arquivos
if((pq=fopen("pr_downlinkG1.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}

if((pq=fopen("pr_downlinkG2.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}

%Computando as métricas
while(g1<=0.2){
    g2=(m1*g1)/(m1*g1 + m2*(1-g1));

    y1=(m1*g1)*pow((1-g1),(m1-1));
    y2=g2;

    n1=pow((1-g1),m1);
    n2=1-g2;

    Pr1=(n1*(y1+y2))/(1+y1+y2);
    Pr2=(n2*(y1+y2))/(1+y1+y2);

    fprintf(pq, " %f ",Pr1);
    fprintf(pq, " %f ",Pr2);

    g1=g1+0.01;
}
fclose(aq);
fclose(pq);
}

```

I.2.6 Probabilidade do evento de downlink caso S-Aloha/NC

```
%Bibliotecas
#include<stdio.h>
#include<stdlib.h>
#include<math.h>

%Função principal
main(){

%Declaração de variáveis
float n1,n2,y1,y2,g1,g2,Pr1,Pr2;
int m1,m2;

%Variável tipo ponteiro para arquivo
FILE *aq,*pq;

%Inicialização de variáveis
m1=5;
m2=1;
g1=0.02;

%Gerando arquivos
if((pq=fopen("pr_downlinkG1NC.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}

if((aq=fopen("pr_downlinkG2NC.txt","a")) == NULL){
    printf("\n 0 arquivo nao pode ser aberto.");
    system("pause");
}

%Computando as métricas
while(g1<=0.2){
    g2=(m1*g1)/(m1*g1 + m2*(1-g1));

    y1=(m1*g1)*pow((1-g1),(m1-1));
    y2=g2;

    n1=pow((1-g1),m1);
    n2=1-g2;

    if(y1>=y2)
        y=y1;
    else
        y=y2;
```

```
Pr1=(n1*(y))/(1+y);
Pr2=(n2*(y))/(1+y);

fprintf(pq, " %f ",Pr1);
fprintf(aq, " %f ",Pr2);

g1=g1+0.01;
}
fclose(aq);
fclose(pq);
}
```