

**ESCOLA POLITÉCNICA - UNIVERSIDADE DE SÃO PAULO**

**DEPARTAMENTO DE ENGENHARIA ELETRÔNICA**

**ÁREA DE SISTEMAS ELETRÔNICOS**

**ANÁLISE DE UM SISTEMA DE COMUNICAÇÃO POR PACOTES**

**PARA USO EM TELEFONIA MÓVEL MICROCELULAR**

Eng. José Augusto de Lima

DEZEMBRO/1996

**ESCOLA POLITÉCNICA - UNIVERSIDADE DE SÃO PAULO**

**DEPARTAMENTO DE ENGENHARIA ELETRÔNICA**

**ÁREA DE SISTEMAS ELETRÔNICOS**

**ANÁLISE DE UM SISTEMA DE COMUNICAÇÃO POR PACOTES**

**PARA USO EM TELEFONIA MÓVEL MICROCELULAR**

Eng. José Augusto de Lima

Dissertação de Mestrado apresentada à  
EPUSP como parte dos requisitos exigidos  
para a obtenção do título de

**MESTRE EM ENGENHARIA ELÉTRICA**

Orientador: Prof. Dr. Paul Jean Etienne  
Jeszensky

DEZEMBRO/1996

Lima, José Augusto de

Análise de um Sistema de Comunicação por Pacotes para Uso em Telefonia Móvel  
Microcelular. São Paulo 1996  
p. 145

Dissertação (Mestrado) - Escola Politécnica da Universidade de São Paulo. Departamento  
de Engenharia Eletrônica -Área de Sistemas Eletrônicos.

1. PRMA 2.Telefonia Móvel 3. Comunicação por Pacotes  
Universidade de São Paulo. Escola Politécnica . Departamento de Engenharia Eletrônica -  
Área de Sistemas Eletrônicos.

## **AGRADECIMENTOS**

Ao meu filho Rafael, privado tanto tempo de minha convivência durante a realização deste trabalho e a minha esposa Vera pela compreensão e incentivo.

Ao amigo Paul Jean Etienne Jeszensky por sua dedicação e que sempre esteve pronto a ceder seu tempo em todos os momentos deste trabalho.

## **RESUMO**

Este trabalho tem como objetivo, analisar o desempenho de um sistema móvel de comunicação de voz e dados acoplados, com transmissão por pacotes PRMA (Packet Reservation Multiple Access).

O índice de desempenho aqui considerado é a probabilidade de perda de pacotes de voz, que quando mantido abaixo de um limiar permite uma comunicação de voz aceitável.

## **ABSTRACT**

The main goal of this work is the performance analysis of PRMA (Packet Reservation Multiple Access). This is a wireless communication system that is able to transmit either package voice or voice and coupled data both of them with voice package reservation.

The main performance measure is the voice package drop probability, because when it is lower than a specific value the voice transmission quality is acceptable.

## **LISTAS DE ALGUMAS ABREVIACÕES UTILIZADAS**

- A/D - Analógica/Digital
- ALOHA - É uma expressão de amor ou boas vindas, em língua Havaiana.
- CDMA - Code Division Multiple Access
- CON - Contending
- D/A - Digital/Analógica
- EMT - Estação Móvel Terminal
- ERB - Estação Rádio Base
- FDMA - Frequency Division Multiple Access
- FIFO - First in First Out
- MAN - Metropolitan Area Network
- MATLAB - Matrix Laboratory (software)
- MFSK - M-Frequency Shifty Keying
- MPSK - M-Phase Shifty Keying
- MQAM - M-Quadrature Amplitude Modulation
- NR - Newton Raphson (algoritmo)
- PAD - Packet Assembler and Disassembler
- PCM - Pulse Code Modulation

- PCN - Personal Communication Network
- PRMA - Packet Reservation Multiple Access
- R-ALOHA - Reservation ALOHA
- TDMA - Time Division Multiple Access
- UIB - Unidade Interface Estação Base
- UIC - Unidade Interface Controle
- UIT - Unidade Interface Central Telefônica
- UIW - Unidade de Interface sem Fio

# ÍNDICE

I - INTRODUÇÃO.....	1
I.1 - OBJETIVO.....	1
I.2 - MOTIVAÇÃO.....	1
I.3 - ESTRUTURA DO TRABALHO.....	2
II - DESCRIÇÃO DO SISTEMA PRMA.....	3
II.1 - ARQUITETURA DO SISTEMA.....	3
II.1.1 - O Canal de Comunicação.....	4
II.1.2 - Arranjo da Rede.....	5
II.1.3 - Reuso do Espectro.....	5
II.1.4 - Os Protocolos ALOHA e ALOHA com "slot".....	5
II.1.5 - Arquitetura da Comutação Celular.....	7
II.2 - CONTROLE DA COMUTAÇÃO REDE CELULAR POR PACOTES.....	9
II.2.1 - Pacotes de Múltiplo Acesso com Reserva (PRMA).....	12
II.3 - CARACTERIZAÇÃO DO DESEMPENHO DE UM SISTEMA PRMA PARA VOZ.....	15
II.3.1 - O Acesso ao Canal e Permissão.....	16
II.3.2 - Modelo para a Voz.....	16
II.3.3 - Detector Lento de Atividade Vocal.....	16
II.3.4 - Detector Rápido de Atividade Vocal.....	17
II.3.5 - Dimensionamento do Tamanho do "Buffer" de Armazenamento.....	18
II.3.6 - Análise do Ponto de Equilíbrio do Sistema.....	19
II.3.7 - Análise para Detector Vocal Lento.....	20
II.3.8 - Equilíbrio do Sistema.....	21
II.3.9 - Análise para Detector Vocal Rápido.....	23
II.3.10 - Estabilidade do PRMA.....	25
II.3.11 - Estimativa da Distribuição de Probabilidade dos Estados.....	26
II.3.12 - Probabilidade de Perda de Pacotes.....	27
II.4 - MODELO COM DADOS ACOPLADOS.....	28
II.4.1 - Modelo de Voz.....	28
II.4.2 - Subsistema de Dados.....	29
II.4.3 - Cálculo do Desempenho pelo Ponto de Equilíbrio para o Modelo com Voz.....	31
III - MÉTODOS DE CÁLCULO PARA SISTEMAS PRMA.....	39
III.1 - SISTEMAS PRMA APENAS COM VOZ.....	39
III.1.1 - Método de Cálculo para a Probabilidade de Perda de Pacotes.....	39



III.2 - SISTEMAS PRMA COM DADOS ACOPLADOS.....	56
III.2.1 - Resultados Obtidos.....	56
III.2.1 - Exemplo de Sistema PRMA.....	63
III.3 - CONCLUSÕES.....	66
IV - REFERÊNCIAS BIBLIOGRÁFICAS.....	67
ANEXO 1      LISTAGEM COMENTADA DAS SUBROTINAS UTILIZADAS.....	69
ANEXO 2      ESTABILIDADE DO PRMA.....	130
ANEXO 3      ESTATÍSTICA DA PERDA DE PACOTES.....	134
ANEXO 4      DESCRIÇÃO DO MÉTODO DE NEWTON RAPHSON (NR) NA RESOLUÇÃO DE UM SISTEMA NÃO LINEAR.....	138

# **I - INTRODUÇÃO**

## **I.1 - OBJETIVO**

Este trabalho tem como objetivo, analisar o desempenho de um sistema móvel de comunicação de voz e dados acoplados, com transmissão por pacotes.

## **I.2 - MOTIVAÇÃO**

Redes integradas constituem um tema de intenso estudo, e crescimento, no mundo atual das telecomunicações. O propósito de uma rede integrada é o de coletar e permitir a troca de informações entre as diversas fontes de uma forma eficiente.

Atualmente os sistemas de telefonia móvel celular em uso comercial no mundo, utilizam acesso por divisão de frequência (FDMA), onde as portadoras são moduladas em FM pela informação. Estes sistemas podem ser classificados como de primeira geração.

Nos EUA e Europa já podemos encontrar sistemas de segunda geração, que utilizam a técnica de multiplexação no tempo (TDMA). Neste caso a informação é digitalizada e este sinal digital modula uma portadora por algum dos tipos de modulação digital convencionais: MFSK, MPSK, MQAM etc. A transmissão é por surtos, em intervalos de tempo alocados especificamente para tal.

Atualmente encontram-se em nível avançado de estudos algumas propostas para o futuro sistema de comunicação pessoal sem fio, conhecido como "Personal Communication Network - PCN". Estes sistemas consistirão de microcelulas, que serão multiplexadas no tempo-TDMA ou por código-CDMA.

Como uma quarta geração pode-se citar estudos preliminares em sistemas, ainda microcelulares, com transmissão de voz e dados, por pacotes com reserva, apoiadas sobre Redes Metropolitanas (MAN), implementadas com fibras ópticas a altas taxas de transmissão. Neste trabalho estuda-se o desempenho de um sistema proposto desta geração segundo as referências [1], [2], [3] e [12].

## **I.3 - ESTRUTURA DO TRABALHO**

O estudo da performance de sistemas PRMA foi realizado em quatro capítulos e quatro anexos. No primeiro capítulo apresenta-se uma introdução do sistema. No segundo capítulo é apresentada uma aplicação prática genérica da utilização do sistema em estudo. Em seguida apresenta-se o modelamento do sistema, o estudo da sua estabilidade e métodos de cálculo da probabilidade de perda de pacotes para sistemas utilizando somente canais de voz com detectores de atividade vocal lentos e rápidos. Neste mesmo capítulo apresentam-se ainda o estudo de sistemas com dados acoplados, os métodos de cálculo utilizados, um possível método de resolução para o problema de instabilidade e a influência dos canais de dados nos resultados de desempenho do sistema. Também é estudado o atraso causado nos canais de dados devido às características adotadas para os canais de voz e dados.

No capítulo três desenvolvem-se os métodos descritos no capítulo dois e que foram implementados em MATLAB, versão 4.0. Neste capítulo fazem-se comparações entre os diversos métodos de modelamento utilizados para sistemas PRMA: só com voz e sistemas com voz e dados acoplados. O capítulo quatro lista as referências bibliográficas utilizadas para este trabalho.

Este trabalho inclui ainda quatro anexos. No primeiro apresenta-se a listagem comentada das subrotinas desenvolvidas para o cálculo de alguns índices de desempenho do sistema descrito. Nos dois anexos seguintes, para o sistema PRMA utilizando somente canais de voz é discutida a sua estabilidade (no anexo dois) e apresenta-se também a estatística de perda de pacotes de voz (no anexo 3). No último anexo apresenta-se o método utilizado para a resolução de um sistema de equações não lineares que foi utilizado para a determinação do ponto de equilíbrio do sistema com dados acoplados.

## II - DESCRIÇÃO DO SISTEMA PRMA

### II. 1 - ARQUITETURA DO SISTEMA

A fig. 2.1 descreve, em diagrama de blocos, a arquitetura do sistema proposto, segundo a ref.[1].

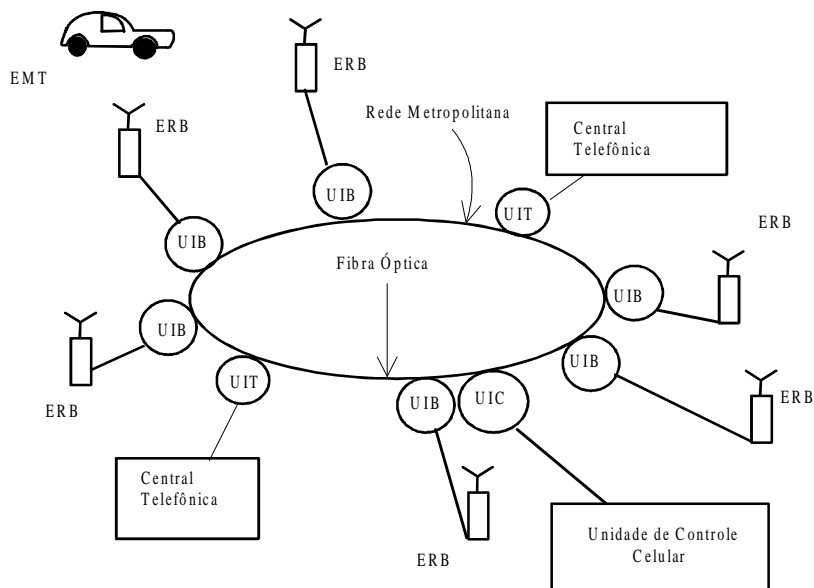


fig. 2.1

onde:

EMT - Estação Terminal Móvel

ERB - Estação Rádio Base

UIB - Unidade Interface Estação Base

UIC - Unidade Interface Controle

UIT - Unidade Interface Central Telefônica

O sistema compreende os elementos essenciais para um acesso sem fio à uma rede de informações fixa, isto é, estações terminais móveis sem fio, estações rádio-base e centrais de comutação.

A rede será totalmente em fibra óptica, utilizando uma taxa de transmissão alta para a comunicação entre estações rádio-base e centrais telefônicas. Utilizará microcélulas com transmissores de baixa potência, afim de facilitar o reuso de frequências, entre outros fatores; o controle do "handoff", por sua vez, será efetuado pela própria estação terminal, abandonando-se assim a prática usual de redes com controle totalmente centralizado.

Como características intrínsecas de uma rede de comunicação móvel desta geração pode-se citar:

- a - os canais de comunicações entre as estações terminais móveis e bases variam com o tempo;
- b - a comunicação é altamente dinâmica, pois uma determinada canalização pode variar muitas vezes por segundo, devido aos movimentos possíveis das estações terminais;
- c - dada a limitação da banda autorizada para as redes sem fio, o reuso eficiente de frequências é essencial.

### **II.1.1 - O Canal de Comunicação**

Como mencionado, o canal de comunicação pode variar muito rapidamente devido ao movimento possível da estação terminal. Raciocinando em termos de pior caso, a SNR (Relação Sinal Ruído) pode piorar a um ponto que torne a comunicação impossível; além deste fator existem mudanças devido a topologia do terreno e interferências (que podem ocorrer aleatoriamente ou serem oriundas de células distantes que utilizam o mesmo grupo de frequências). Devido a esses fatores, esta rede deverá, em princípio, usar na comunicação entre estação terminal e rádio-base, uma taxa baixa de transmissão, ref.[5], suficiente para o que se propõe; um tipo de modulação robusta; códigos corretores de erro, refs.[4] e [6]; mudança dinâmica na frequência de transmissão e recepção ("frequency hopping"), ref.[7]; diversidade de espaço, ref.[8]; equalização não linear, refs.[9] e [10], entre outros fatores.

### **II.1.2 - Arranjo da Rede**

No sistema celular, as transmissões estão confinadas a pequenas áreas que são chamadas de células. Assim, um usuário poderá iniciar a sua transmissão em uma determinada célula K e ir trocando de célula, durante a transmissão da informação. Atualmente o controle do sistema é centralizado, mas com o aumento da demanda deste serviço haverá a necessidade de se usar, cada vez mais, células menores e assim o controle não poderá mais ser totalmente centralizado.

### **II.1.3 - Reuso do Espectro**

Devido ao fato da banda autorizada para este sistema ser pequena e de que no futuro ocorrerá um estrangulamento dos sistemas já existentes, os estudos atuais dirigem-se para sistemas microcelulares. Os sistemas de primeira geração usam células com raio de aproximadamente 10 Km, para uma banda autorizada pequena. Assim a densidade de tráfego que se pode escoar é menor que  $1E/MHz/Km^2$ . Com uma banda permitida de aproximadamente 50MHz, em pouco tempo necessitar-se-á aumentar esta densidade por um fator de aproximadamente 10, devido a demanda por novos serviços de transmissão de voz, dados e imagem, e isto implicará na necessidade de reuso mais eficiente das frequências e na busca de métodos mais eficientes para a transmissão/multiplexação.

### **II.1.4 - Os Protocolos ALOHA e ALOHA com "slot"**

Uma das formas de caracterizar protocolos de multiacesso pode ser, por exemplo, de acordo com o tipo de controle que é realizado. Quando dois terminais disputarem o mesmo "slot" e ocorrer uma colisão, uma nova tentativa de transmissão poderá ser estabelecida, sob certas condições. Conforme estas condições de sistema, variarão: a vazão do sistema ("Throughput"), que é a relação entre a quantidade de mensagens bem sucedidas e o total, por unidade de tempo; o atraso do sistema etc. O desempenho do sistema pode ser medido, analisando-se estas características.

Os protocolos considerados podem ainda ser com ou sem reserva. É dito que um protocolo tem reserva quando tão logo as chamadas vão sendo recebidas elas entram em uma fila tipo FIFO ("First In- First Out"). Haverão perdas ("drop"), quando todas as memórias FIFO estiverem

preenchidas. Neste caso, uma estratégia possível será eliminar aquela informação que estiver armazenada a mais tempo.

O protocolo ALOHA foi desenvolvido na década de 70, para permitir a comunicação entre o Centro de Computação da Universidade do Havaí e seus terminais, geograficamente dispersos. Neste protocolo cada terminal, assim que receber a mensagem da fonte, tentará transmiti-la imediatamente. Se houver colisão com a transmissão de um outro terminal, cada terminal envolvido na colisão retransmitirá sua mensagem e, para evitar-se repetidas colisões dos mesmos terminais, cada retransmissão será feita em tempos aleatórios. O terminal detecta se sua transmissão colidiu, ou não, através de uma informação da Central, via um outro canal auxiliar. A fig. 2.2 ilustra a colisão de pacotes.

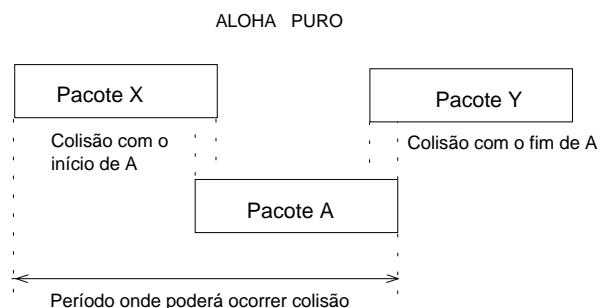


fig. 2.2

Roberts, na ref.[11], descreve o sistema ALOHA com "slots" de comprimentos constantes e com sincronismo na transmissão, de forma a diminuir o período vulnerável pela metade. Este conceito está ilustrado na fig. 2.3.

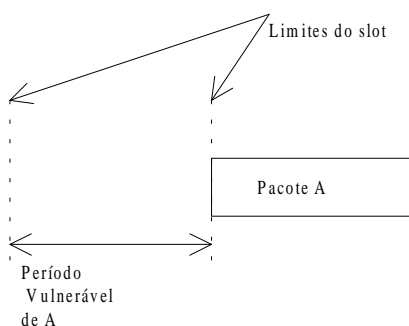


fig. 2.3

### II.1.5 - Arquitetura da Comutação Celular

Sistemas com uma arquitetura como a aqui descrita distribuem o controle da rede entre pequenos processadores, denominados de Unidades de Interfaces, que são residentes em todos elementos da rede. Na fig. 2.1 ilustra-se a infraestrutura de uma rede metropolitana (MAN). A informação entra e sai da MAN através de unidades de interfaces celulares; neste caso: Unidades de Interfaces de Estações Bases (UIB), Unidades de Interfaces de Centrais (UIT), Unidade de Interface de Controle (UIC) e as Unidades de Interface dos Terminais sem fio (IUW). Como se verificará ao longo deste trabalho, a capacidade de comunicação/comutação de uma rede MAN é elevada para protocolos com reserva (PRMA), na transferência de informações entre as ERB's e os terminais móveis sem fio.

As interfaces UIW, UIB, UIT e UIC organizam a transferência da informação entre os terminais sem fio, estações bases, centrais telefônicas e controlador celular. Cada pacote será identificado por um remetente e um destinatário. Descrevem-se a seguir, resumidamente, as funções de cada interface.

#### a - Unidade de Interface de Centrais (UIT)

Estas interfaces aceitam, e autorizam, a saída de informações em um formato padrão, por exemplo 64 Kbps em PCM ("Pulse Code Modulation").



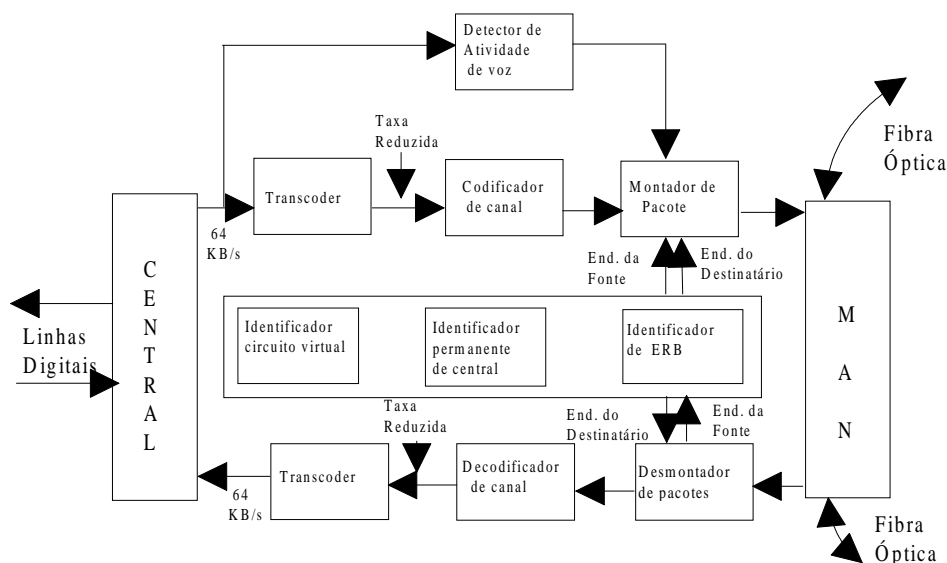


fig. 2.4

O desmontador de pacotes (PAD), lê o endereço de destino em todos pacotes que estão transitando e o montador, por sua vez, insere o do terminal no pacote a ser enviado.

O detector de atividade de voz, controla a geração dos pacotes pelo PAD, para que durante intervalos de silêncio, não se gerem pacotes.

#### **b - Unidade de Interface para Terminal sem fio (UIW)**

Na geração de pacotes as funções são similares às da UIT. Uma diferença básica é que a UIT recebe as informações digitais da rede pública, enquanto a UIW está conectada à uma fonte analógica de informações, necessitando pois de conversores A/D e D/A. Como indicado na fig. 2.5, o montador de pacotes na UIW envia seus pacotes para o rádio transmissor, sob o controle de um processador de protocolos PRMA. Para implementar uma troca de célula ("handoff") iniciado pelo terminal, a UIW consulta o circuito de monitoração da qualidade dos canais. Este circuito escolhe então a melhor ERB para a situação atual.

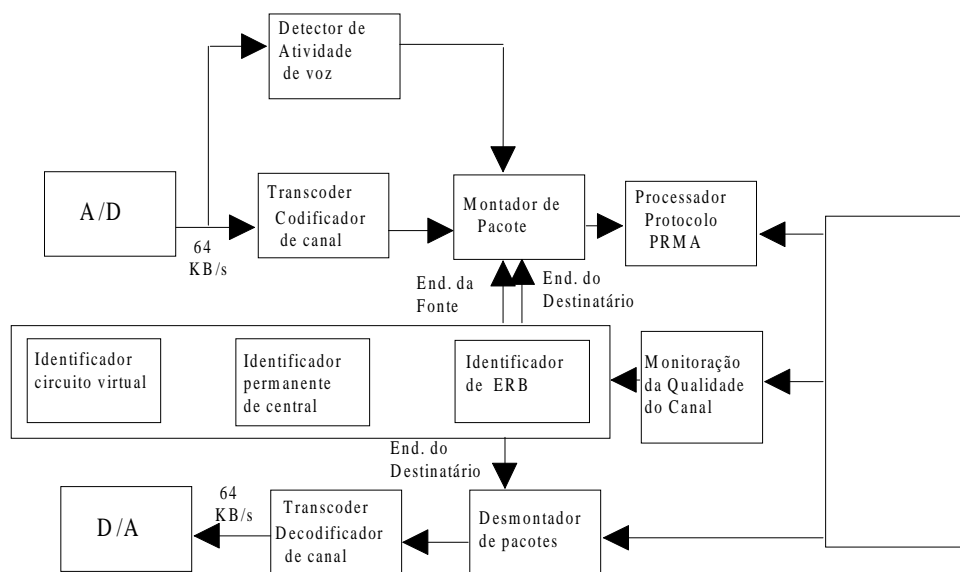


fig. 2.5

### c - Unidade de Interface de Estação Base

Esta unidade conecta as informações entre UIT's e as estações terminais sem fio.

### d - Unidade de Interface de Controle Celular

O controle celular recebe, processa e gera o controle dos pacotes na rede. Ele é sempre endereçado pelo seu identificador permanente (número fictício dado para as chamadas).

## II.2 - CONTROLE DA COMUTAÇÃO NA REDE CELULAR POR PACOTES, REF.[1]

Em cada pacote, existem o endereço do remetente e do destinatário, afim de controlar o fluxo de dados para a unidade de interface correta. Ao iniciar-se uma chamada, o controlador celular fornece a esta chamada um número identificador virtual. Este número relacionará univocamente os endereços da UIT e da UIW envolvidas na chamada. Estações bases e controladores celulares são sempre endereçados pelo seu identificador permanente.

### a - Conversação sem troca de célula ("handoff")

Enquanto o veículo encontrar-se na mesma célula, o pacote transita no sentido do terminal, passando pela ERB e daí para a central; no outro sentido a transmissão é na direção oposta, de uma maneira direta, sem troca do identificador virtual (número recebido pela ligação que identifica a chamada). Na fig. 2.6, que ilustra o processo, pode-se observar que não houve troca de ERB.

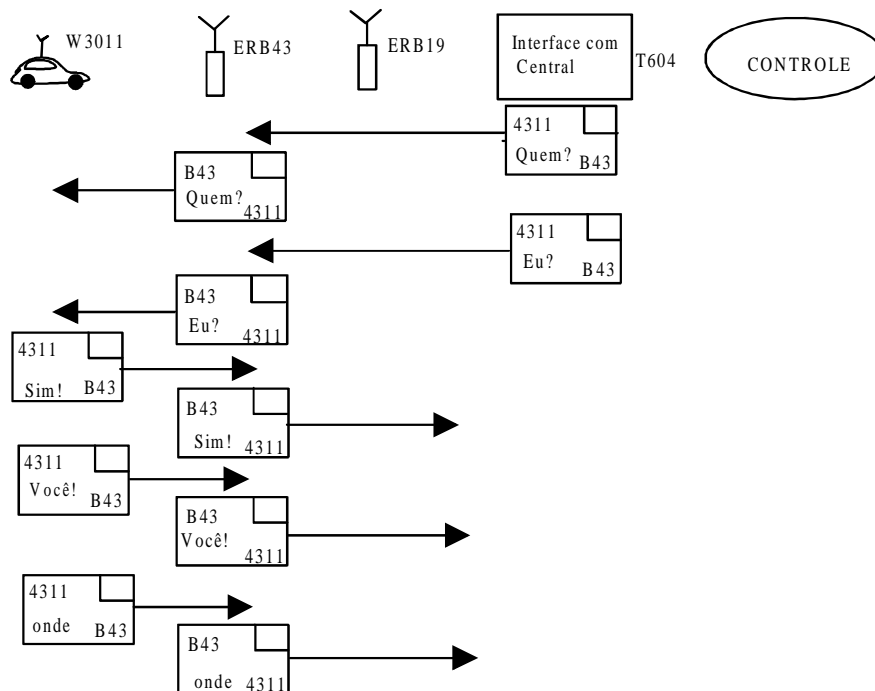


fig. 2.6

### b - Conversação com troca de célula ("Handoff")

Neste caso o pacote muda de destino de uma ERB para outra, mantendo porém o mesmo identificador virtual e quem faz esta troca é a estação móvel. Observe-se no exemplo da fig. 2.7 a seguir, que a ERB mudou de B43 para B19.

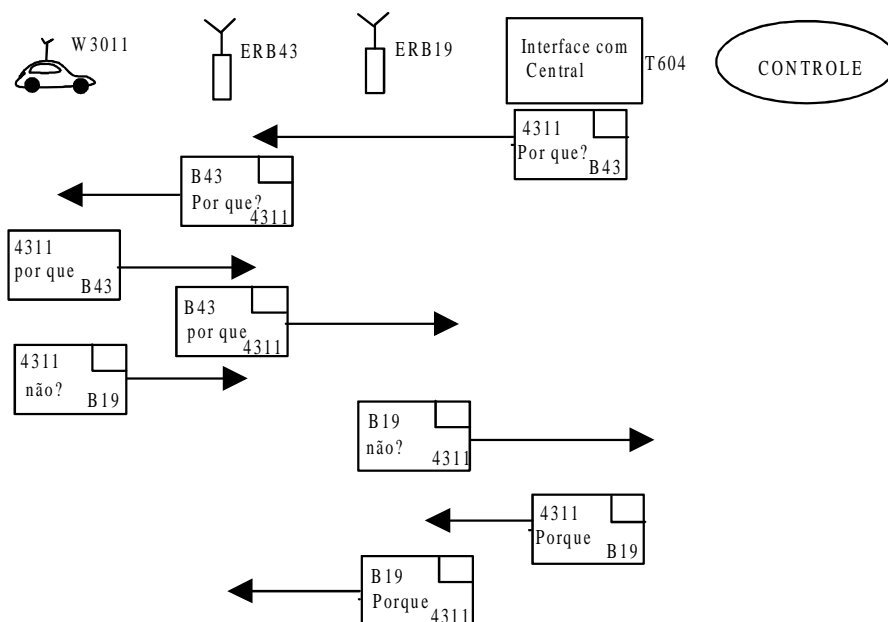


fig. 2.7

### c - A Estação Terminal inicia uma chamada ("Call Set-up")

A fig. 2.8 ilustra um possível caso de início de chamada. Observe-se que no início de uma chamada a estação terminal W3011, conecta-se com o controle através da estação rádio-base B43, afim de receber o tom de discar. O controle autoriza, através de B43, o envio do tom de discar ao terminal solicitante. A estação terminal discar e recebe o número de identificação da chamada, neste caso 4311, e daí em diante a chamada continua com este número.

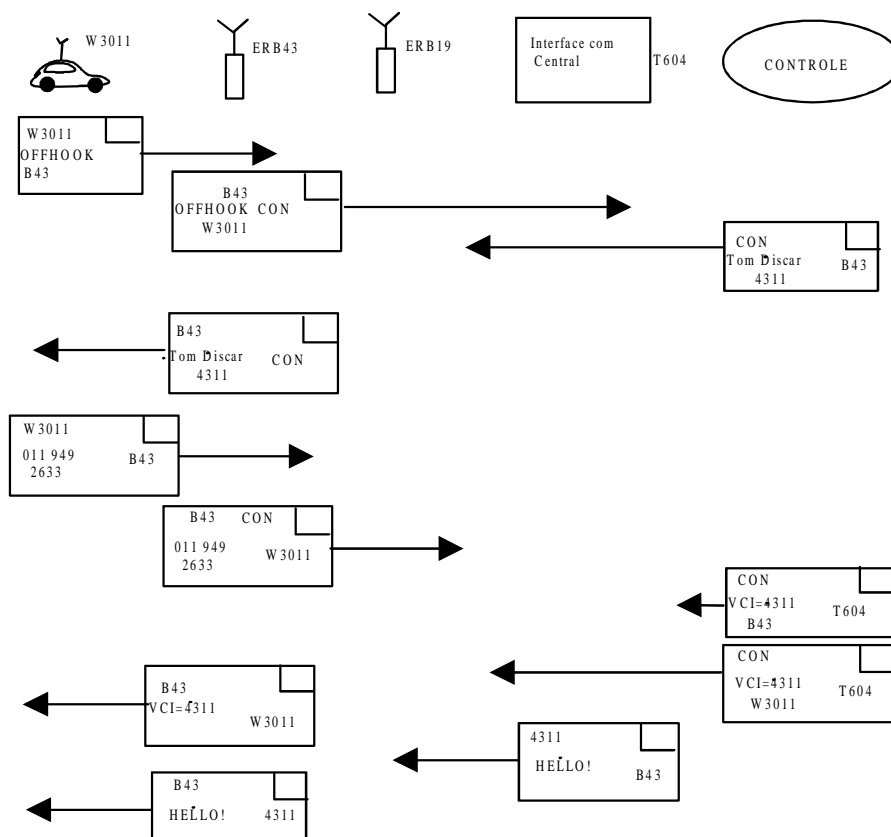


fig. 2.8

## II.2.1 - Pacotes de Múltiplo Acesso com Reserva (PRMA)

Analisa-se inicialmente o protocolo PRMA sem dados e a seguir com os dados acoplados.

Até este ponto foi descrita a composição do sistema, e seu funcionamento, em termos de comutação. Far-se-á agora uma breve descrição do pacote, para o qual será determinado o desempenho.

PRMA é um processo de transmissão/multiplexação semelhante ao R-ALOHA. Ele se distingue do R-ALOHA por sua resposta ao congestionamento da rede e pelos seus curtos intervalos de transmissão. No R-ALOHA um congestionamento pode causar grandes atrasos na transmissão dos pacotes; no caso PRMA estas informações são descartadas, caso elas estejam esperando por um intervalo de tempo maior que um valor previamente especificado. O canal de comunicação é

organizado em intervalos de tempo, denominados de "slots", de tal modo que cada "slot" pode carregar informações dos terminais para as ERB's. Os "slots" são agrupados em "frames" e as estações terminais são capazes de, dentro de um "frame", reconhecer se cada "slot" está reservado ou não, tomando como base o "frame" anterior proveniente da ERB. Para as estações terminais, assim como no caso ALOHA, as que tiverem novas informações a transmitir entrarão no estado de disputa. No final de cada "slot", a ERB enviará uma indicação se a transmissão (disputa) foi bem sucedida, isto é, obteve-se reserva ou não. A duração do "frame" é escolhida de forma que o terminal gere um pacote por "frame". Após ganhar uma reserva, o terminal continua a usar seu "slot" até não ter mais informação a transmitir. A ERB que não recebe pacotes em um determinado "slot" reservado previamente, avisa a todos os terminais que este "slot" está livre para ser reservado e/ou disputado. O mecanismo de permissão é baseado na probabilidade de permissão  $p$ , um parâmetro constante de definição do sistema, que é a probabilidade de se transmitir um pacote para a ERB em um "slot" disponível. Uma vantagem do protocolo PRMA é a de utilizar os instantes de silêncio entre surtos de voz de uma conversação para transmitir outro canal; isto faz com que o ganho do sistema aumente. A fig. 2.9 descreve, com um exemplo, o funcionamento do protocolo PRMA.

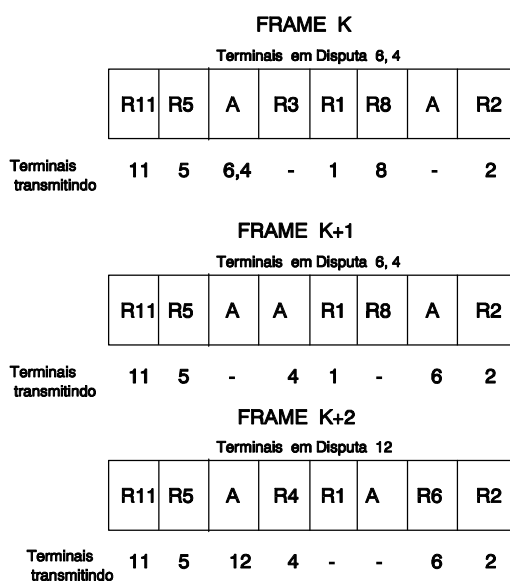


fig. 2.9

"frame" K

Observe-se que as estações terminais 6 e 4 disputam um mesmo "slot" livre, apesar de existirem dois livres, isto é, estão em conflito.

"frame" K+1

As estações 6 e 4 disputam agora "slots" diferentes e a estação 3 parou de transmitir.

"frame" K+2

As estações 6 e 4 receberam autorização de uso das reservas solicitadas, a estação 12 disputa um "slot" livre e a estação 8 parou de enviar informação.

O número de "slots" disponíveis em um determinado instante depende do número de terminais em comunicação efetiva (estado de envio de informação). Quando existirem vários terminais ocupados, observa-se que o tempo de espera para se conseguir uma reserva aumenta. No caso do protocolo PRMA define-se um intervalo de tempo máximo para espera,  $D_{max}$  (na prática implementado por um "buffer" tipo FIFO), e neste trabalho adotar-se-à o valor de 32ms, a menos que conste explicitamente outro. Afim de não necessitar-se de controle de eco, quando usado em rede de telefonia pública, este valor delimita o tempo de espera a partir do qual, caso não se consiga uma reserva, o terminal descartará a sua informação mais antiga. Para se avaliar o desempenho do protocolo PRMA, pode-se estudar a probabilidade de descarte,  $P_{drop}$ . Para estimar a capacidade do sistema admite-se uma  $P_{drop} \leq \epsilon$ , em geral 1%, e com este valor pode-se determinar o número máximo de terminais que compartilham o sistema, dentro deste limite. A fig. 2.10 descreve uma possível estrutura interna para o protocolo PRMA.

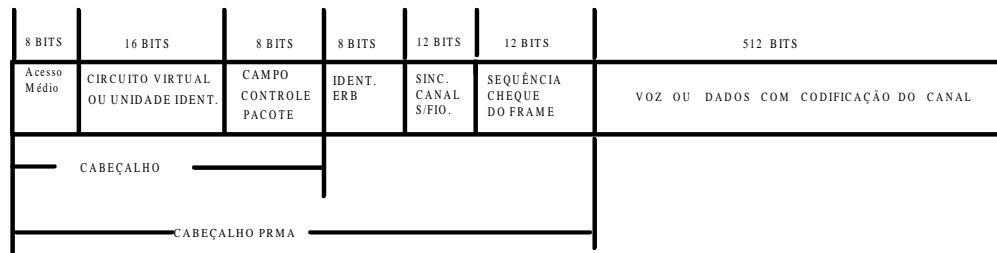


fig. 2.10

## II.3 - CARACTERIZAÇÃO DO DESEMPENHO DE UM SISTEMA PRMA PARA VOZ, REF.[2]

Sejam então:

$R_s$  - taxa da fonte (informação);

$R_c$  - taxa do canal;

$H$  - número de "bits" do cabeçalho;

$T$  - duração do "frame";

$N$  - número de "slots" por "frame".

A duração  $T$  do "frame" é uma variável de projeto e a estrutura deste é projetada de forma que as fontes de voz gerem, exatamente, um pacote por "frame". Nestas circunstâncias tem-se:

$$N = \left\lceil \frac{R_c \cdot T}{R_s \cdot T + H} \right\rceil \text{ pacotes/"frame"} \quad (2.1)$$

onde a notação  $[x]$  indica a parte inteira de  $x$ . A duração do "slot" é:

$$\tau = T/N \quad (2.2)$$



### II.3.1 - O Acesso ao Canal e Permissão

O canal PRMA é composto por "slots" que são agrupados formando um "frame". A taxa de "frame" é dada pela taxa de pacotes. No caso em estudo admite-se que a probabilidade de permissão é idêntica para todas estações móveis e independente da SNR, significando que quando há disputa de um "slot" por duas estações terminais, não há uma diferenciação na escolha do terminal por SNR.

### II.3.2 - Modelo para a Voz

No primeiro caso modela-se o sistema somente para voz e existirá assim um detector de atividade vocal que criará um padrão de "terminal falando" e "terminal em silêncio". Se este detector de atividade vocal for rápido ele conseguirá detectar não somente os surtos de voz ("spurts") e silêncio, como também minisurtos ("minispurts") e minisilêncios ("minigaps"), devidos aos curtos espaços de tempo de silêncio que são característicos de uma conversa.

### II.3.3 - Detector Lento de Atividade Vocal

A fig. 2.11 descreve este modelo que é um processo de Markov de dois estados.

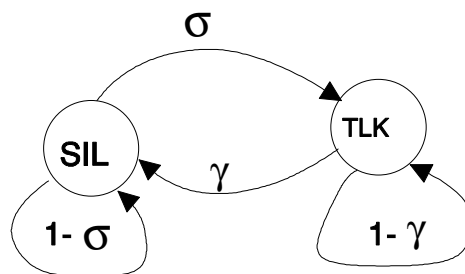


fig. 2.11

A probabilidade de que um surto de voz, com duração média  $t_1$  s., termine em um "slot", de duração  $\tau$  s., é dada por:

$$\gamma = 1 - e^{-\tau/t_1} \quad (2.3)$$

A probabilidade de que haja uma transição do estado falando (TLK) para o de silêncio (SIL) corresponde a probabilidade de que um intervalo de silêncio, de duração média  $t_2$  s., termine em um "slot", de duração  $\tau$  s., e é dada por:

$$\sigma = 1 - e^{-\tau/t_2} \quad (2.4)$$

### II.3.4 - Detector Rápido de Atividade Vocal

Neste caso tem-se três estados possíveis: Falando (TLK), intervalo de silêncio principal (SILp) e minisilêncio (SILm).

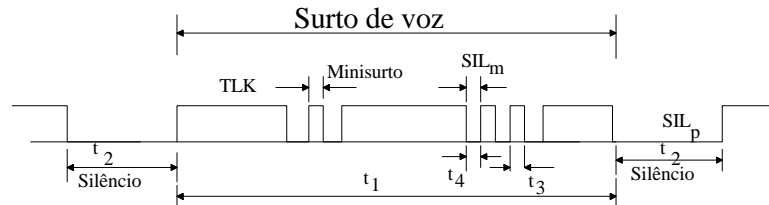


fig. 2.12

A probabilidade de que haja uma transição do estado SILp para o de TLK durante um "slot" de tempo, de duração  $\tau$  s., é  $\sigma$ , tal como na equação (2.4). Minisilêncios têm duração média  $t_4$  s. e a probabilidade de haver uma transição de SILm para TLK é, de forma análoga às anteriores, dada por:

$$\sigma_m = 1 - e^{-\tau/t_4} \quad (2.5)$$

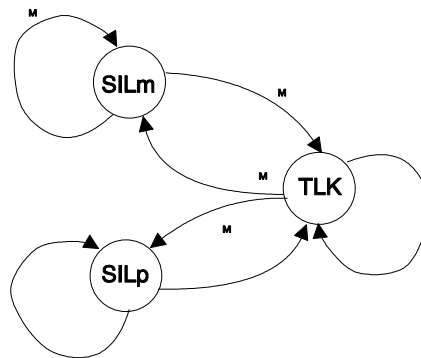


fig 2.13

No final de um minisurto de voz, de duração média  $t_3$  s., existe uma transição de TLK para SILm, dado que este não seja o último num surto de voz principal. A probabilidade de que um minisurto termine em um "slot" de tempo, de duração  $\tau$  s., é dada por:

$$\gamma_m = 1 - e^{-\tau/t_3} \quad (2.6)$$

O número médio de minisurtos, a cada surto principal, é dada pela razão  $\frac{t_1}{(t_3 + t_4)}$  e desta forma:

$$\alpha = \frac{(t_3 + t_4)}{t_1} \quad (2.7)$$

é a probabilidade de que um minisurto seja o último no seu surto principal e com isso a probabilidade de haver uma transição de um estado TLK para SILp é  $\alpha\gamma_m$ .

A duração esperada do minisurto e do minisilêncio são características do detector de atividade vocal. Com  $t_4=0$  (não houve deteção de minisilêncio) e  $t_1=t_3$  (não há distinção entre minisurtos e surto principal), tem-se  $\alpha = 1$ , o que mostra que o detector lento pode ser considerado como um caso especial do rápido.

### II.3.5 - Dimensionamento do Tamanho do "Buffer" de Armazenamento

Um terminal de voz contém uma memória, tipo FIFO, para armazenamento de pacotes que estão esperando permissão para transmissão. Se a capacidade do "buffer" for de B pacotes e estiver preenchido totalmente, quando chegar um novo pacote no terminal este "buffer" eliminará o primeiro pacote (armazenado a mais tempo) e armazenará o novo. Com este mecanismo pode-se escrever para B:

$$B = \lceil D_{\max}/T \rceil \quad (2.8)$$

onde:

T - duração do "frame"

$D_{\text{máx}}$  - indica o máximo atraso permitido para a voz na transmissão.

Pode-se definir também D, em unidades de "slots" de tempo, como sendo  $D = \left\lceil \frac{D_{\text{máx}}}{\tau} \right\rceil$  "slots"

### II.3.6 - Análise do Ponto de Equilíbrio do Sistema

A finalidade da análise do ponto de equilíbrio do sistema é a de determinar a influência dos parâmetros do modelo de voz e das variáveis do PRMA sobre o comportamento do sistema, ref.[13]. Para analisar o desempenho do sistema, modela-se-o como um processo de Markov. Seja N o número de pacotes PRMA por "frame"; então cada estação terminal poderá estar em um dos (N+3) estados descritos abaixo.

$$\{SIL_m, SIL_p, CON, RES_0, \dots, RES_{N-1}\} \quad (2.9)$$

$SIL_m$  - minisilêncio;

$SIL_p$  - silêncio principal;

CON - disputa;

$RES_i$  - mantendo uma reserva para o i-ésimo futuro "slot".

O estado  $\Omega$  do sistema é uma configuração de (N+3) variáveis definidas como o número de terminais em cada um dos estados de terminais, definidos por (2.9).

$$\Omega = \{S_m, S_p, C, R_0, \dots, R_{N-1}\} \quad (2.10)$$

Para estudar o sistema em estado estacionário, deve-se encontrar a probabilidade de transição de qualquer um dos estados para todos os outros possíveis e determinar assim  $\Phi(\Omega)$ , que é a distribuição de probabilidade assintótica de  $\Omega$ . A probabilidade de perda de pacote, assim como outras medidas de desempenho estatístico interessantes, são esperanças com respeito a  $\Phi(\Omega)$ . A análise precisa do processo de Markov, neste caso (N+3) dimensional, é muito complexa e faz-se

então, alternativamente, uma análise pelo ponto de equilíbrio. Desta forma chega-se a uma configuração de variáveis de sistema, referidas como pontos de equilíbrio, que são os valores de equilíbrio dos elementos de  $\Omega$ , definidos em (2.10).

$$\omega = \{s_m, s_p, c, r_1 \dots r_{N-1}\} \quad (2.11)$$

No equilíbrio ter-se-á:  $C = c, S_m = s_m, S_p = s_p$  e  $R_i = r_i$  com  $0 \leq i \leq N$ , e a esperança da taxa com que os terminais deixam um determinado estado é a mesma com que os terminais entram neste mesmo estado.

É possível ter-se sistemas com múltiplos pontos de equilíbrio e, evidentemente, isto indica que os valores de equilíbrio são distintos dos valores esperados para estas variáveis. Este caso será analisado no item II.3.10

### II.3.7 - Análise para Detector Vocal Lento

Apresentam-se a seguir as características essenciais para a análise de um detector de voz lento. O diagrama de estados e transições do terminal é o representado na fig. 2.14 abaixo.

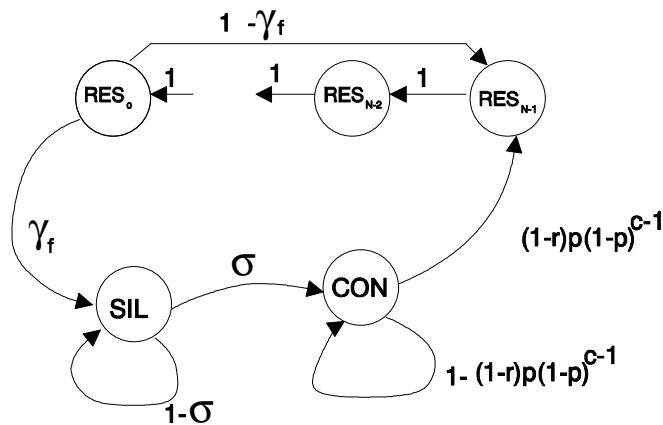


fig. 2.14

A fig. 2.14 representa um terminal de voz PRMA com  $(N+2)$  estados, isto é, o detector de voz reconhece somente um estado de silêncio, denominado SIL. As transições ocorrem somente ao final de cada "slot". Pela figura pode-se observar que, estando o terminal no estado SIL, ele vai para o de CON (disputa) se houver a ocorrência de um surto de voz, com probabilidade  $\sigma$ , em cada "slot". Do estado  $RES_i$ ,  $i \neq 0$ , o terminal sempre vai para o estado  $RES_{i-1}$ . Para que haja transição do estado  $RES_0$  para o de  $RES_{N-1}$  o terminal deverá estar armazenando pacotes para transmissão. Após transmitir o último pacote, o terminal vai do estado  $RES_0$  para o SIL, com probabilidade  $\gamma_f$  que é a probabilidade de que o surto termine no "frame" mais recente. Esta probabilidade é dada por:

$$\gamma_f = 1 - (1 - \gamma)^N \approx N\gamma \quad (2.12)$$

onde  $\gamma$  é a probabilidade de que o surto de voz termine em um dos  $N$  "slots" no "frame".

Para sair do estado CON (disputa) e ir para o estado  $RES_{N-1}$  (reserva), o terminal deve obedecer às três condições seguintes:

- a - Haja um "slot" disponível;
- b - Tenha permissão para transmitir e
- c - Não haja nenhuma colisão com um pacote de outro terminal .

As permissões ocorrem independentemente em todos os terminais e são independentes dos estados de reserva do canal. Assim as três condições para se obter uma reserva são independentes e a probabilidade de ocorrer uma transição de CON para  $RES_{N-1}$ , na figura anterior, é um produto de três termos.

### II.3.8 - Equilíbrio do Sistema

No ponto de equilíbrio tem-se:

$$c = eq\{C\} = \text{número de terminais em equilíbrio no estado CON}$$

$s = \text{eq}\{S\} =$  número de terminais em equilíbrio no estado SIL

$r_i = \text{eq}\{R_i\} =$  número de terminais em equilíbrio no estado  $\text{RES}_i$

Note-se que no ponto de equilíbrio todos os "slots" têm iguais probabilidades,  $r$ , para serem reservados, podendo-se portanto escrever:

$$r_0 = r_1 = r_2 = \dots = r_{N-1} = r \quad (2.13)$$

Esta igualdade decorre do equacionamento dos fluxos de entrada e saída nos estados de transição  $\text{RES}_{N-2}, \dots, \text{RES}_0$ . No equilíbrio a probabilidade de transição do estado CON para o estado  $\text{RES}_{N-1}$  é dado pelo produto das probabilidades:

$(1-r)$  - probabilidade de um "slot" disponível.

$p$  - probabilidade de permissão (parâmetro do sistema).

$u(c)$  - probabilidade de acesso não contestada, que corresponde à probabilidade de nenhum dos outros terminais em disputa ter permissão para transmitir no "slot" corrente.

Demonstra-se, na ref. [2], que:

$$u(c) = \begin{cases} (1-p)^{c-1} & p/c \geq 1 \\ 1 & p/c < 1 \end{cases} \quad (2.14)$$

Observando-se o estado RES no "buffer" N-1 tem-se:

$$r(1-\gamma_f) + \text{cpu}(1-r)(1-\gamma) - r = 0 \quad (2.15)$$

No estado SIL tem-se:

$$s\sigma = r\gamma_f + c\gamma \quad (2.16)$$

O número total de terminais de voz é M, então:

$$s + c + Nr = M \quad (2.17)$$

Obtém-se assim um sistema com três equações a três incógnitas  $\mathbf{c}$ ,  $\mathbf{s}$  e  $\mathbf{r}$ .

$$\begin{cases} r(1-\gamma_f) + \mathbf{c}p(1-r)(1-\gamma) = r \\ r\gamma_f + \gamma\mathbf{c} = s\sigma \\ s + \mathbf{c} + N\mathbf{r} = M \end{cases} \quad (2.18)$$

Onde, após eliminar-se  $\mathbf{r}$  e  $\mathbf{s}$ , obtém-se:

$$\mathbf{c} \left( 1 + \frac{\gamma}{\sigma} \right) + r \left( N + \frac{\gamma_f}{\sigma} \right) = M \quad (2.19)$$

Como  $\mathbf{u}$  é uma função de  $\mathbf{c}$ , a equação (2.19) tem uma única incógnita  $\mathbf{c}$ , que é o número de terminais em disputa no estado de equilíbrio. Cada solução de (2.19) é um valor de equilíbrio de  $\mathbf{C}$  e o número de valores de equilíbrio está relacionado com a estabilidade do sistema, como se verá adiante.

### II.3.9 - Análise para Detector Vocal Rápido

Apresentam-se a seguir as características essenciais para a análise de um detector de voz rápido. O diagrama de estados e transições do terminal é o a seguir representado pela fig. 2.15.

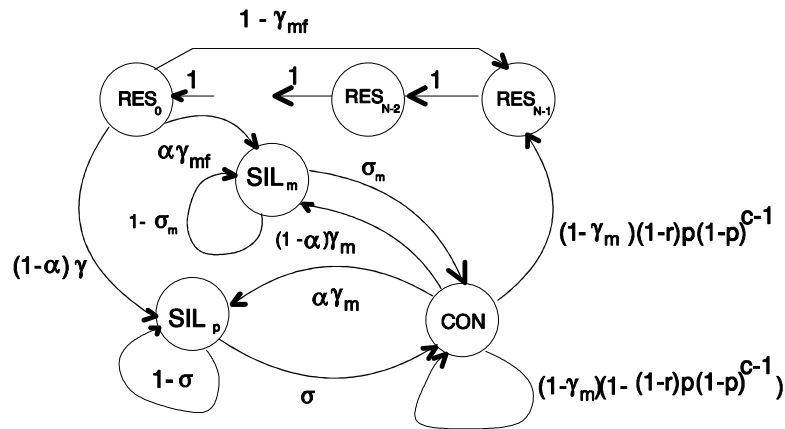


fig. 2.15



Esta figura estende-se em relação à 2.14, em dois aspectos. Ela incorpora o modelo geral de atividade vocal e mostra as transições do estado de disputa (CON) para o de silêncio principal (SIL) e para o de minisilêncio (SIL<sub>m</sub>). Estas transições implicam que se um surto de voz terminar antes que o terminal obtenha uma reserva, este parará a disputa. Seguindo a convenção adotada anteriormente, denota-se por  $S_m$  e  $S_p$  o número de terminais no estado SIL<sub>m</sub> e SIL<sub>p</sub>, respectivamente. Os valores de equilíbrio correspondentes são designados por  $s_m$  e  $s_p$ .

Exceto para  $\gamma_{mf}$ , todos os símbolos são idênticos aos já introduzidos nas figuras anteriores;  $\gamma_{mf}$  é análogo à  $\gamma_f$  da equação (2.12) e corresponde à probabilidade que um minisurto de voz termine no "frame" mais recente.

$$\gamma_{mf} = 1 - (1 - \gamma_m)^N \approx N\gamma_m \quad (2.20)$$

Neste caso as equações correspondentes ao sistema (análogas ao sistema 2.18) são:

$$\begin{cases} s_m + s_p + c + Nr = M \\ r(1 - \gamma_{mf}) + cpu(1 - r)(1 - \gamma_m) = r \\ r\gamma_{mf}(1 - \alpha) + c\gamma_m(1 - \alpha) - s_m\sigma_m = 0 \\ r\gamma_{mf}\alpha + c\gamma_m\alpha - s_p\sigma = 0 \end{cases} \quad (2.21)$$

Onde, após eliminar-se  $s_m$ ,  $s_p$  e  $r$ , obtém-se:

$$c \left[ 1 + \frac{(1 - \alpha)\gamma_m}{\sigma_m} + \frac{\alpha\gamma_m}{\sigma} \right] + \frac{cpu(1 - \gamma_m)}{cpu(1 - \gamma_m) + \gamma_{mf}} \left[ N + \frac{\alpha\gamma_{mf}}{\sigma} + \frac{(1 - \alpha)\gamma_{mf}}{\sigma_m} \right] = M \quad (2.22)$$

Denotando-se o lado esquerdo de (2.22) por  $F(c)$ , denominada de Função de Equilíbrio do Sistema, a solução escreve-se:

$$F(c) = M \quad (2.23)$$

Observa-se que (2.23) tem um número ímpar de soluções, para valores de  $c$  entre 0 e  $M$ , pois  $F(0)=0$  e  $F(M)>M$ .

### II.3.10 - Estabilidade do PRMA

Define-se como ponto de equilíbrio estável do sistema aquele estado donde, se houver alguma pequena perturbação das variáveis, o sistema é levado de volta para o mesmo ponto anterior (de equilíbrio). Para um ponto de equilíbrio instável, uma pequena perturbação das variáveis de estado leva-o para um ponto distinto do inicial. Para estudar a estabilidade do PRMA, define-se a função  $G(c)$ , onde  $c$  é o valor de equilíbrio do número de terminais em disputa.

$$G(c) = c\gamma_m + cpu(1 - \gamma_m)(1 - r) \quad (2.24)$$

$G(c)$  corresponde ao fluxo de saída do estado de disputa, no equilíbrio. A estabilidade dos pontos de equilíbrio pode ser caracterizada pelas derivadas de  $G(c)$ . Se houver um aumento de  $c$  lentamente, no ponto de equilíbrio, devido a um aumento na entrada do estado de disputa, obtém-se  $G'(c) > 0$ . Isto implica que o fluxo de saída  $G(c)$  de disputa também aumenta. Um decréscimo em  $c$  leva a um decréscimo em  $G(c)$ . Desta forma  $G'(c) > 0$  implica que o ponto de equilíbrio é estável e, com raciocínio análogo,  $G'(c) < 0$  que o ponto de equilíbrio é instável.

Na ref.[2] demonstra-se que o "throughput" do sistema PRMA,  $\eta$ , é exatamente a probabilidade de reserva de "slot" no equilíbrio,  $r$ , e que este "throughput" tem um máximo quando  $r'(c) = 0$ . Denotando-se este valor máximo de  $c$  por  $c_0$  demonstra-se no anexo 2, que:

$$c_0 = \frac{-1}{\ln(1 - p)} \quad (2.25)$$

para  $c > c_0$ , o "throughput" decresce com o aumento do tráfego oferecido e a perda de pacotes aumenta rapidamente. Diz-se que o sistema está congestionado.

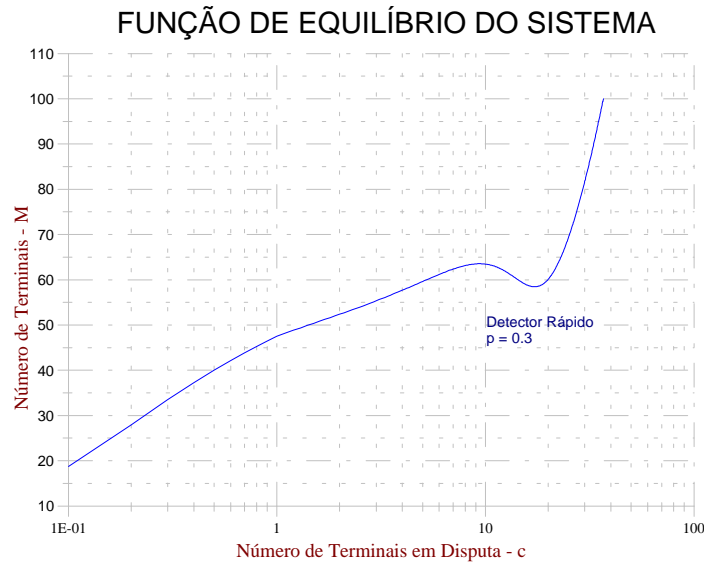


fig. 2.16

Mostra-se ainda no mesmo anexo, que para um ponto de equilíbrio estável deve-se ter  $c < c_0$  e  $G'(c) > 0$  (estes podem ser chamados de "bons pontos" para operação de um sistema PRMA corretamente projetado).

### II.3.11 - Estimativa da Distribuição de Probabilidade dos Estados

A probabilidade de perda de pacotes pode ser avaliada se forem conhecidas as distribuições correspondentes às variáveis aleatórias em questão.

Desde que os terminais atuem independentemente e sem sincronismo, pode-se assumir que  $R_0, \dots, R_{N-1}$  são independentes e identicamente distribuídas (iid). Define-se então uma nova variável,  $R$ , como sendo a soma das  $N$  variáveis iid  $R_i$ .

$$R = \sum_{i=0}^{N-1} R_i \quad (2.26)$$

É fácil verificar que  $R$  corresponde ao número de "slots" reservados num "frame". Dado que a probabilidade de reserva no equilíbrio é  $r$ , conclui-se que a distribuição de  $R$  é binomial, de forma que a probabilidade de  $R$  "slots" estarem reservados dentre  $N$ , pode-se escrever:

$$\Theta_R = \binom{N}{R} r^R (1-r)^{N-R} \quad (2.27)$$

Pode-se agora determinar a distribuição como função de  $R$  e  $C$ , onde  $C$  é o número de terminais em disputa, a partir de:

$$\Theta(C, R) = \Theta_{C/R}(C/R) \Theta_R(R) \quad (2.28)$$

e de hipóteses adicionais para  $\Theta(C, R)$ . Na ref.[2] adota-se, levando em conta que  $C+R \leq M$ :

$$\Theta_{C/R}(C/R) = \begin{cases} p_0(1-p_0)^C & C < M-R \\ (1-p_0)^{M-R} & C = M-R \\ 0 & \text{para outros valores} \end{cases} \quad (2.29)$$

$$\text{com } p_0 = \frac{1}{c+1} \quad (2.30)$$

pela constatação de que o estado de disputa é análogo a uma "fila de serviço" para a obtenção de reserva, o que sugere que o número de elementos na fila é geometricamente distribuído.

### II.3.12-Probabilidade de Perda de Pacotes

Um terminal PRMA elimina todos os seus pacotes que precisem esperar mais do que  $D$  "slots" para obter uma reserva. Para determinar a perda de pacotes deve-se derivar a probabilidade de que um terminal obtenha reserva para  $j$  "slots", após o início de um surto de voz, com  $j$  inteiro. Não haverá perdas de pacotes se  $j \leq D$ . Se  $j > D$  um pacote será perdido, mais um para cada "frame" adicional ( $N$  "slots") que o terminal tenha que esperar para obter a reserva. Para o modelo simplificado obtém-se, ver anexo 2:

$$P_{\text{drop}} = \gamma_f \frac{v^D}{1 - (1 - \gamma_f)v^N} \quad (2.31)$$

onde:

$$v = v(C, R) = \left[ 1 - (1 - R/N)p(1 - p)^C \right] \quad (2.32)$$

é a probabilidade de que um terminal em disputa não obtenha uma reserva no "slot" corrente.

Demonstra-se, ainda na mesma referência anterior, que para o modelo com detectores rápidos de voz, tem-se:

$$P_{\text{drop}}(v) = \frac{v^D \gamma_{mf} (1 - \gamma_{mf})^{B-1}}{1 - v^N} \times \left\{ 1 - \frac{\gamma_{mf} [1 - (1 - \gamma_{mf})v^{2N}]}{[1 - (1 - \gamma_f)v^N]^2} \right\} + \frac{\gamma_{mf}^2 v^N}{[1 - (1 - \gamma_{mf})v^N]} + \gamma_{mf}^2 (1 - \gamma_{mf})^{B-1} (v^D - v^{BN}) \quad (2.33)$$

onde:

$$v = v(C, R) = 1 - (1 - R/N)p(1 - p)^C (1 - \gamma_m) \quad (2.34)$$

Em seguida pode-se calcular a probabilidade de perda incondicional como a esperança sobre a distribuição estimada de  $\Theta(C, R)$ :

$$P_{\text{drop}} = \sum_{R=0}^{N-1} \sum_{C=0}^{M-R} \Theta_R(R) \cdot \Theta_{C/R}(C/R) \cdot P_{\text{drop}}(v) \quad (2.35)$$

## II.4 - MODELO COM DADOS ACOPLADOS

### II.4.1 - Modelo de Voz

No modelo com dados acoplados foi adotado na análise do desempenho, os modelos de voz com detectores rápidos e lentos. A fig.2.17 mostra o sistema adotado para voz com detectores lentos.

Este difere do sistema da fig. 2.14, devido ao fato de que neste caso adota-se que um terminal em disputa poderá voltar ao estado de silêncio com probabilidade  $\gamma$ .

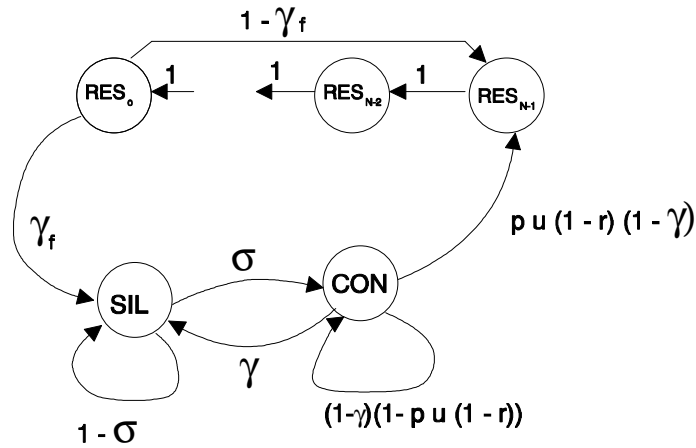


fig. 2.17

#### II.4.2 - Subsistema de Dados

Neste sub-sistema adota-se, que a probabilidade de um pacote ser gerado por um terminal de dados, em qualquer "slot" é  $\sigma_d$ . Pode-se calcular a taxa média de dados transmitidos por um terminal como sendo:

$$R_d = \sigma_d R_s N \text{ bps} \quad (2.36)$$

$R_d \Rightarrow$  Taxa média da transmissão de dados.

Sejam ainda:

$pd \Rightarrow$  Probabilidade de permissão para dados.

$b \Rightarrow$  Número de terminais de dados no equilíbrio.

Assume-se que os terminais de dados não têm reserva, isto significa, que a cada transmissão, estes devem disputar um "slot" livre. Desta forma as memórias de armazenamento, "buffers" do pacotes de dados, são suficientemente grandes afim de que os terminais de dados não descartem informações, que já permaneceram armazenadas por um determinado intervalo de tempo. Um

terminal com dados à transmitir (TDT), chamado de carregado, pode transmitir um pacote num "slot" não reservado se ele tiver permissão para transmitir.

Um TDT obtém sucesso na transmissão em um determinado "slot", se assegurarem-se as seguintes condições:

- 1 - O "slot" não está reservado.
- 2 - O terminal tem permissão com probabilidade **pd**.
- 3 - Nenhum outro terminal de voz tem permissão.
- 4 - Nenhum outro TDT tem permissão.

Não se considera o caso onde os terminais de dados podem obter reservas pois os mesmos podem ter uma quantidade de dados muito grande à transmitir, e isto degradaria as características da probabilidade de perda de pacotes dos sistema de voz, aumentando por conseguinte o congestionamento do mesmo. Estudos do congestionamento de sistemas com dados acoplados que obtêm reserva de "slots", é um tópico a ser estudado no futuro.

Um terminal que já tem **j** pacotes na sua memória, está no estado BUFj, e o projeto do sistema é tal que não mais do que um pacote proveniente de um terminal chega a um mesmo "slot". Para cada terminal de dados adota-se o processo de vida-morte mostrado abaixo.

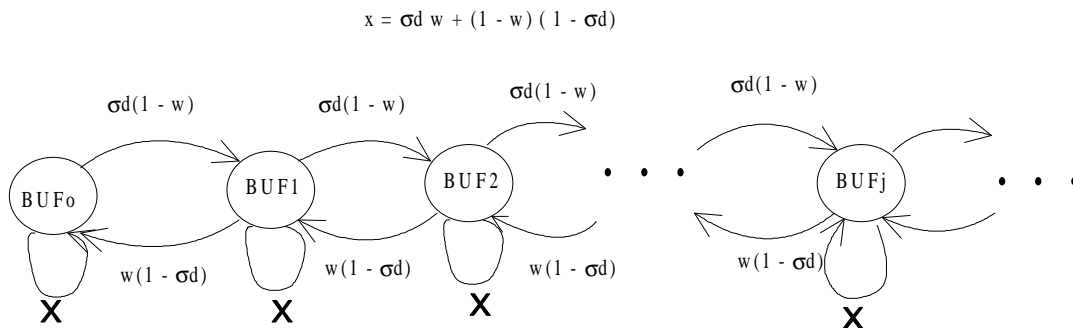


fig. 2.18

onde:

$\omega_d \sigma_d + (1 - \omega)(1 - \sigma_d)$ : representa a probabilidade de gerar um novo pacote e conseguir-se transmitir **ou** não se gerar nenhum e não se conseguir transmitir o anterior

$\omega(1 - \sigma_d)$ : representa a probabilidade de conseguir a transmissão e não se gerar nenhum pacote

$\sigma_d(1 - \omega)$ : representa a probabilidade de gerar um novo pacote e não houve sucesso na transmissão anterior.

### II.4.3 - Cálculo do Desempenho pelo Ponto de Equilíbrio para o Modelo com Voz

#### II.4.3.1 - Método de Cálculo, segundo Nanda ref.[12]

##### a - Equacionamento para a voz

De maneira análoga a que chegou-se em (2.18) tem-se:

$$\begin{cases} r\gamma_f + c\gamma - s\sigma = 0 \\ r(1 - \gamma_f) + cpu(1 - r)(1 - \gamma) - r = 0 \\ s + c + Nr = M \end{cases} \quad (2.37)$$

onde **u** é dado por:

$$u = (1 - p)^{c-1} (1 - p_b)^b \quad (2.38)$$

Neste caso **u** é uma função não linear de **b** e **c**. Na ausência de terminais de dados esta é uma equação em **c** e pode-se escrever:

$$h_1 c + h_2 r = M$$

sendo



$$h_1 = 1 + \frac{\gamma}{\sigma}$$

$$h_2 = N + \frac{\gamma_f}{\sigma}$$

com  $r$  dado pela equação:

$$r = \frac{cpu(1-\gamma)}{cpu(1-\gamma) + \gamma_f} \quad (2.39)$$

e onde o número total de terminais de voz  $M$  é dado por:

$$c \left( 1 + \frac{\gamma}{\sigma} \right) + r \left( N + \frac{\gamma_f}{\sigma} \right) = M \quad (2.40)$$

### **b - Equacionamento para dados**

Na análise pelo ponto de equilíbrio para um subsistema de dados, a probabilidade de equilíbrio  $w$  de uma transmissão de dados com sucesso é dada por:

$$\omega = p_d u_d (1-r) \quad (2.41)$$

com  $u_d$  dado por:

$$u_d = (1-p)^c (1-p_d)^b \quad (2.42)$$

Como o número de TDT's não pode exceder ao número total de terminais de dados,  $M_d$ , demonstra-se que:

$$b = \min \left\{ \frac{c}{P} \sigma_d M_d \frac{1-\gamma}{\gamma_f} \left[ \frac{h_2}{M - h_1 c} \right], M_d \right\} \quad (2.43)$$

com  $P$  dado por (2.44)

$$P = \frac{p_d(1-p)}{p(1-p_d)} \quad (2.44)$$

### II.4.3.2 - Método de Cálculo Proposto

#### a -Modelo com voz e dados

O modelo de voz adotado é o da fig. 2.17 e o modelo para dados é o da fig. 2.18. Resolvendo-se chega-se ao sistema de equações dado por (2.45).

$$\left\{ \begin{array}{l} c \left( 1 + \frac{\gamma}{\sigma} \right) + r \left( M + \frac{\gamma_f}{\sigma} \right) = M \\ r = \frac{cpu (1-r)}{cpu (1-\gamma) + \gamma_f} \\ b u_d p_d (1-r) = s_d M_d \end{array} \right. \quad (2.45)$$

onde  $\mathbf{u} = f(p, p_d, b, c)$  e  $\mathbf{ud} = g(p, p_d, b, c)$ , variam de acordo com a região considerada.

As incógnitas são:  $\mathbf{b}$ ,  $\mathbf{c}$ , e  $\mathbf{r}$ . Observa-se que este sistema pode ser reduzido a duas equações com duas incógnitas  $\mathbf{b}$  e  $\mathbf{c}$ . Encontra-se o valor do par ordenado  $(b, c)$ , que é a solução do sistema para um dado número de terminais de voz  $\mathbf{M}$ , um dado número de terminais de dados  $\mathbf{Md}$ , uma probabilidade de acesso de dados  $\mathbf{pd}$  e uma probabilidade de acesso de voz  $\mathbf{p}$ . Ao tentar resolver o problema acima vê-se claramente que as equações tornam-se complexas pois, na maioria dos casos, não existe um desacoplamento entre  $\mathbf{b}$  e  $\mathbf{c}$ . A técnica adotada neste caso, foi dividir o plano  $(b, c)$  em quatro regiões, tal como mostra a fig. 2.19, e resolver-se o sistema de equações em cada região, separadamente.

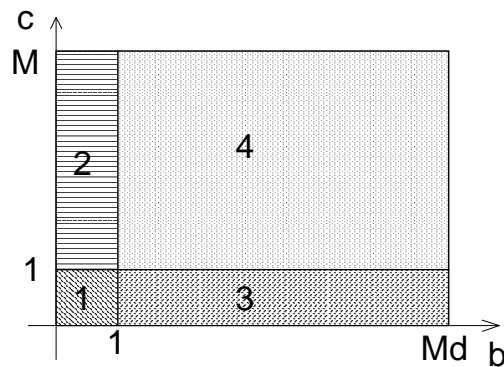


fig 2.19

### b - Soluções a serem calculadas por intervalos

O sistema dado na forma geral em (2.45), deve ser resolvido nas hipóteses (regiões) listadas abaixo.

$$\left\{ \begin{array}{l} c \geq 1 \Rightarrow u = (1-p)^{c-1} (1-p_d)^b \end{array} \right. \quad (2.46a)$$

$$\left\{ \begin{array}{l} c < 1 \Rightarrow u = (1-p_d)^b \end{array} \right. \quad (2.46b)$$

$$\left\{ \begin{array}{l} b \geq 1 \Rightarrow u_d = (1-p)^c (1-p_d)^{b-1} \end{array} \right. \quad (2.46c)$$

$$\left\{ \begin{array}{l} b < 1 \Rightarrow u_d = (1-p)^c \end{array} \right. \quad (2.46d)$$

$$\left\{ \begin{array}{l} c < 1 \wedge b < 1 \text{ usar (2.46b) e (2.46d)} \rightarrow \text{região (1)} \\ c \geq 1 \wedge b < 1 \text{ usar (2.46a) e (2.46d)} \rightarrow \text{região (2)} \\ c < 1 \wedge b \geq 1 \text{ usar (2.46b) e (2.46c)} \rightarrow \text{região (3)} \\ c \geq 1 \wedge b \geq 1 \text{ usar (2.46a) e (2.46c)} \rightarrow \text{região (4)} \end{array} \right.$$

Observação: nas hipóteses (2.46a) e (2.46c), que corresponde à região 4 da fig. 2.19, tem-se:

$$P = \frac{p_d u_d}{pu} = \frac{p_d (1-p)^c (1-p_d)^{b-1}}{p(1-p)^{c-1} (1-p_d)^b} = \frac{p_d (1-p)}{p(1-p_d)} \quad (2.47)$$

Observa-se que neste caso, e somente neste, este parâmetro **P** independe de **b** e **c** e pode ser utilizado para simplificar a resolução do problema, segundo a ref.[12]. Afim de manter um padrão único de resolução abandona-se o método adotado na ref.[12], que só serviria para a região número 4, e adota-se o mesmo procedimento de cálculo para qualquer região obedecendo suas restrições.

Para resolver o sistema de equações, isola-se **r** e assim pode-se escrever:

$$\left\{ \begin{array}{l} ch_1 + \frac{cpu(1-\gamma)}{cpu(1-\gamma) + \gamma_f} h_2 = M \\ b u_d \frac{P_d}{\sigma_d} \frac{\gamma_f}{cpu(1-\gamma) + \gamma_f} = M_d \end{array} \right. \quad (2.48)$$

Deve-se observar que  $F(b,c)-M=0$  e  $\Phi(b,c)-Md=0$ , ver equação 2.49 adiante, representam cada uma, uma curva no plano  $b \times c$  como se evidencia na fig. 2.20.

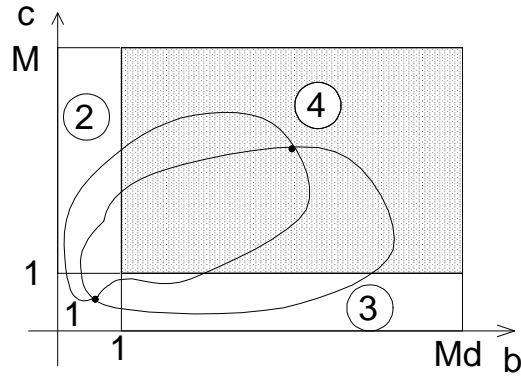


fig 2.20

### c - Equacionamento para cada região

Como para cada região tem-se as probabilidades de acesso incondicionais  $u$  e  $ud$ , para dados e voz, diferentes deve-se ter os quatro casos dados abaixo. Mas independentemente de região pode-se escrever:

$$\begin{cases} f(b,c) = ch_1 + r \cdot h_2 - M \\ \varphi(b,c) = b u_d k_d \cdot (1-r) - M_d \end{cases} \quad (2.49)$$

#### Região 1

Nesta região deve-se resolver o sistema (2.49), preservando apenas as soluções onde  $0 < b < 1$  e  $0 < c < 1$ , dentro das condições dadas por (2.50):

$$\begin{cases} u = (1-p_d)^b \\ ud = (1-p)^c \end{cases} \quad (2.50)$$

#### Região 2

Nesta região deve-se resolver o sistema (2.49), preservando apenas as soluções  $0 < b < 1$  e  $1 < c < M$  dentro das condições dadas por (2.51):

$$\begin{cases} u = (1 - p_d)^b \cdot (1 - p)^{c-1} \\ u_d = (1 - p)^c \end{cases} \quad (2.51)$$

### Região 3

Nesta região deve-se resolver o sistema (2.49), preservando apenas as soluções  $1 < b < Md$

e  $1 < c < M$  dentro das condições dadas por (2.52):

$$\begin{cases} u = (1 - p_d)^b \\ u_d = (1 - p)^c \cdot (1 - p_d)^{b-1} \end{cases} \quad (2.52)$$

### Região 4

Nesta região deve-se resolver o sistema (2.49), preservando apenas as soluções  $1 < b < Md$

e  $1 < c < M$  dentro das condições dadas por (2.53):

$$\begin{cases} u = (1 - p_d)^b \cdot (1 - p)^{c-1} \\ u_d = (1 - p)^c \cdot (1 - p_d)^{b-1} \end{cases} \quad (2.53)$$

## II.4.3.3 - Cálculo da Probabilidade de Perda de Pacotes ( $P_{drop}$ )

Uma vez que já dispõe-se dos valores do número de terminais de voz em disputa no equilíbrio ( $c$ ) e do número de terminais de dados no equilíbrio  $b$ , pode-se optar em calcular a probabilidade de perda de pacotes utilizando distribuições de probabilidades ou resolver pelo ponto de equilíbrio.

### a - Cálculo da probabilidade, $P_{drop}$ , utilizando distribuições de probabilidades

Neste caso substitui-se o valor do número de terminais em disputa  $c$  na equação (2.28) e calcula-se  $\theta(C, R)$ ; a partir deste pode-se calcular  $\Theta(C/R)$  através da expressão (2.29);  $P(v)$  pode ser calculada pela expressão (2.33) e (2.34) e finalmente pode-se obter  $P_{drop}$  pela equação (2.35).

### b - Cálculo da probabilidade, $P_{drop}$ , pelo ponto de equilíbrio

Neste caso, segundo a ref.[12], pode-se calcular  $P_{\text{drop}}(v)$  por (2.54) e (2.55).

$$P_{\text{drop}}(v) = \frac{v^D \gamma_f (1 - \gamma_f)^{B-1}}{1 - v^N} \times \left\{ 1 - \frac{\gamma_f [1 - (1 - \gamma_f) v^{2N}]}{[1 - (1 - \gamma_f) v^N]^2} \right\} + \frac{\gamma_f^2 v^N}{[1 - (1 - \gamma_f) v^N]} + \gamma_f^2 (1 - \gamma_f)^{B-1} (v^D - v^{BN}) \quad (2.54)$$

onde:

$$v = v(C, R) = (1 - \gamma) [1 - (1 - r)p(1 - p)^c] \quad (2.55)$$

#### II.4.3.4 - Atraso de Pacotes de Dados na Transmissão

Neste item estuda-se o atraso médio de pacotes no equilíbrio, para conseguir uma transmissão com sucesso num terminal de dados. Pode-se tomar o tempo médio de espera para uma transmissão com sucesso de um terminal de dados como sendo  $\tau/\omega$ , onde  $\omega$  é probabilidade de um terminal de dados transmitir com sucesso um determinado pacote. Para um pacote que chega num terminal que já tem  $j$  pacotes na fila (estado BUFj), o tempo médio de espera será:

$$\tau(j+1)/\omega.$$

O tempo médio de espera para os pacotes de dados no equilíbrio pode ser calculado por:

$$W_{av} = \tau \sum_{j=0}^{\infty} \frac{j+1}{\omega} P_r(\text{BUF}j) \quad (2.56)$$

onde  $P_r(\text{BUF}j) = \frac{b_j}{M_d}$  representa a probabilidade de um terminal de dados encontrar-se no estado

BUFj. Na ref.[12] simplifica-se esta expressão, cujo resultado é:

$$W_{av} = \frac{\tau}{\omega(1 - \rho_d)} \frac{(1 - \sigma_d)^2}{(1 - \omega)} \cong \frac{\tau}{\omega(1 - \rho_d)} \quad (2.57)$$

onde  $\rho_d$  é a relação entre o número médio de terminais de dados no equilíbrio e o número total de terminais de dados do sistema ( $M_d$ ).

$$\rho_d = \frac{b}{M_d} \quad (2.58)$$

Pode-se adotar limites para o cálculo do tempo médio de espera  $W_{av}$ . Sejam  $W_\beta$  e  $\beta$  tais que:

$$P_r(W \geq W_\beta) \leq \beta \quad (2.59)$$

onde  $W$  é o atraso esperado para um dado pacote chegando num determinado terminal.

Demonstra-se, ainda na ref.[12], que este tempo de espera pode ser encontrado pela equação:

$$W_\beta > \frac{\tau}{\omega} \left( 1 + \frac{\ln \beta}{\ln \rho_d} \right) \quad (2.60)$$

Desta forma tomando-se  $\beta$  igual a 0.05 ou 0.01, por exemplo, pode-se obter o atraso esperado para 95% ou 99% dos pacotes, respectivamente.

### **III - MÉTODOS DE CÁLCULO PARA SISTEMAS PRMA.**

#### **III.1 - SISTEMAS PRMA APENAS COM VOZ**

Neste item evidenciam-se os resultados obtidos com os vários tipos de sistemas descritos no capítulo II.

a - Sistema com detector rápido de atividade voz.

b - Sistema com detector lento de atividade de voz.

Para cada detector foi calculada a probabilidade de perda de pacotes por dois métodos:

1 - Método completo utilizando distribuições de probabilidades.

2 - Método simplificado utilizando as equações de estado no ponto de equilíbrio.

##### **III.1.1 - Método de Cálculo para a Probabilidade de Perda de Pacotes**

###### **III.1.1.1 - Cálculo pelo ponto de Equilíbrio**

Pode-se adotar os seguintes passos para o cálculo da probabilidade de perda de pacotes pelo método do ponto de equilíbrio.

###### **Passo 1:**

Entrada de dados:

De uma forma geral, para todos os cálculos posteriores, utilizar-se-á os dados das tabelas 3.01 e 3.02, descritas a seguir.



**Modelo de voz - Tabela 3.01**

Variável	Símbolo	Valor adotado
Duração média do Surto	t1	1,000 segundos
Duração média do intervalo de silêncio	t2	1,350 segundos
Duração média do minisurto de voz	t3	0,275 segundos
Duração média do minisilêncio	t4	0,050 segundos

**Sistema PRMA - Tabela 3.02**

Variável	Símbolo	Valor adotado
Taxa de transmissão do Canal	Rc	720Kbps
Taxa de transmissão do canal de voz	Rs	32Kbps
Duração do "frame"	T	16ms
Número de "slots" por "frame"	N	20
Duração do "slot"	$\tau$	0,8ms
Tamanho do pacote		576bits
Cabeçalho	H	64 bits
Atraso máximo permissível de voz	Dmáx	32ms
Tamanho do "Buffer" de voz	B	2 pacotes
Probabilidade de permissão para voz	p	variável
Número de terminais de voz	M	variável

**Passo 2:**

De posse dos dados das tabelas 3.01 e 3.02, calcula-se o conjunto de variáveis  $S$  da equação (3.01).

$$S = \{\gamma, \sigma, \sigma_m, \gamma_m, \alpha, N, \tau, B, D\} \quad (3.01)$$

**Passo 3:**

A partir das equações de equilíbrio (2.18) e (2.21), calcula-se o valor do conjunto de variáveis  $\omega$

onde  $\omega = \{c, r, u\}$ .

**c:** número de terminais em disputa (calculado a partir de (2.18) ou (2.21)) dependendo do tipo de sistema.

**u:** calculável a partir de **c** e **p**

**r:** probabilidade de um terminal estar reservado no equilíbrio.

Resolvendo as equações no ponto de equilíbrio chega-se a:

$$M = F(c) \quad (3.02)$$

Observa-se pela fig. 3.01 que o número de soluções desta equação é ímpar e pode-se ter uma ou três soluções.

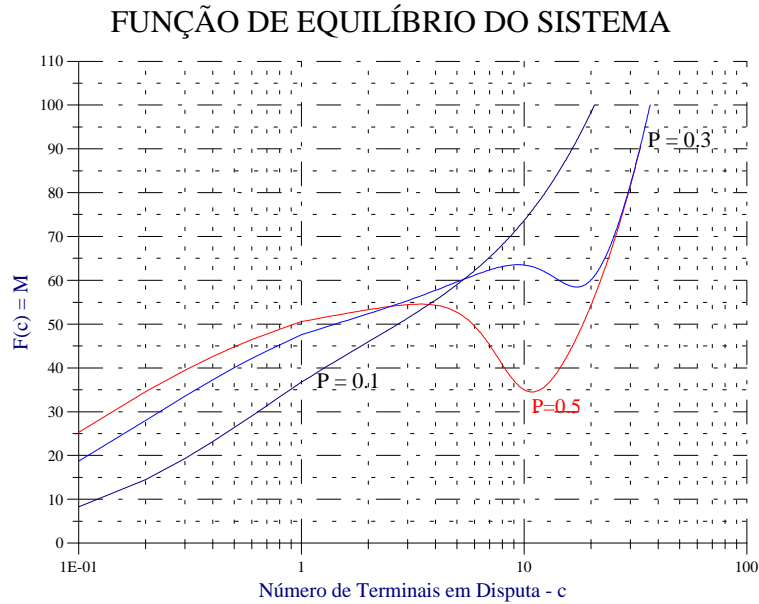


Fig. 3.01

**Passo 4:**

Calcula-se o valor de  $v(c, r)$  que é a probabilidade de um terminal em disputa não conseguir reserva e que é dada por:

- para o modelo com detectores de voz lentos

$$v = (1 - \gamma) \left[ 1 - (1 - r)p(1 - p)^c \right] \quad (3.03)$$

- para o modelo com detectores de voz rápidos

$$v = \left[ 1 - (1 - \gamma_m)(1 - r)p(1 - p)^c \right] \quad (3.04)$$

**Passo 5:**

Cálculo da probabilidade de perda  $P_{\text{drop}}$ .

- para sistemas com detectores lentos

$$P_{\text{drop}} = \frac{v^D \gamma_f (1 - \gamma_f)^{B-1}}{1 - v^N} \times \left\{ 1 - \frac{\gamma_f [1 - (1 - \gamma_f) v^{2N}]}{[1 - (1 - \gamma_f) v^N]^2} \right\} + \frac{\gamma_f^2 v^N}{[1 - (1 - \gamma_f) v^N]} + \gamma_f^2 (1 - \gamma_f)^{B-1} (v^D - v^{BN}) \quad (3.05)$$

- para sistemas com detectores rápidos

$$P_{\text{drop}} = \frac{v^D \gamma_{mf} (1 - \gamma_{mf})^{B-1}}{1 - v^N} \times \left\{ 1 - \frac{\gamma_{mf} [1 - (1 - \gamma_{mf}) v^{2N}]}{[1 - (1 - \gamma_{mf}) v^N]^2} \right\} + \frac{\gamma_{mf}^2 v^N}{[1 - (1 - \gamma_{mf}) v^N]} + \gamma_{mf}^2 (1 - \gamma_{mf})^{B-1} (v^D - v^{BN}) \quad (3.06)$$

### III.1.1.2 - Cálculo Utilizando Distribuições de Probabilidade

Neste método para o cálculo da probabilidade de perda de pacotes utilizam-se distribuições de probabilidade das várias variáveis aleatórias. Na fig 3.02 exemplifica-se um possível "frame" do sistema considerado, onde:

A: "slot" livre para disputa.

R<sub>i</sub>: "slot" reservado para o usuário i.

R: número de "slots" reservados num "frame"

"FRAME" K

R11	R5	A	R1	R2	A	R3	R4
-----	----	---	----	----	---	----	----

fig. 3.02

(no exemplo deste caso R=6)

Os três primeiros passos deste processo são os mesmos discutidos no item III.1.1.1. De posse dos dados das tabelas 3.01 e 3.02, calcula-se o conjunto de variáveis **S** dados por:

$$S = \{\Theta(R), \Theta(C/R), V(C, R), P_{drop}(C, R)\} \quad (3.07)$$

#### **a - Cálculo de $\Theta(R)$**

Sendo **r**: probabilidade de reserva no equilíbrio calcula-se:

$$\Theta_R(R) = \binom{N}{R} r^R (1-r)^{N-R} \quad (3.08)$$

A probabilidade de todos "slots" estarem reservados ocorre quando  $R=N$ , logo:

$$\Theta_R(N) = r^N \quad (3.09)$$

$\Theta_R(R)$ : é a probabilidade de  $R$  "slots" estarem reservados dentre os  $N$  possíveis.

Para  $R=1$  tem-se:

$$\Theta_R(1) = Nr(1-r)^{N-1} \quad (3.10)$$

#### **b - Cálculo de $\Theta(C/R)$**

$$\Theta(C, R) = \Theta_{C/R} \left( \frac{C}{R} \right) \Theta_R(R) \text{ pela regra de Bayes} \quad (3.11)$$

O primeiro membro relaciona-se com a probabilidade de perda de pacote. Como  $C$  é o número de terminais em disputa

$$C + R \leq M \quad (3.12)$$

(na fig. 3.02 tem-se  $R=6$ ,  $M$ =não definido e  $C=2$ )

$\Theta(C, R)$  é a distribuição para o número de terminais em disputa (descondicionado de  $R$ )

Observa-se que com o aumento de R, diminui-se o número possível para C e a expressão (3.11) fica descondicionada. Por razões físicas adota-se:

$$\Theta_{C/R}(C, R) = \begin{cases} p_o(1-p_o)^C & C < M - R \\ (1-p_o)^{M-R} & C = M - R \\ 0 & \text{outros valores} \end{cases} \quad (3.13)$$

com

$$p_o = \frac{1}{c+1}$$

#### **c - Cálculo de $v(C, R)$**

$v(C, R)$  é a distribuição de probabilidade de C terminais estarem em disputa e não conseguirem reserva. Esta distribuição é dada por:

- para o modelo com detectores lentos

$$v(C, R) = (1 - \gamma) [1 - (1 - R/N)p(1-p)^C] \quad (3.14)$$

- para o modelo com detectores rápidos.

$$v(C, R) = [1 - (1 - \gamma_m)(1 - R/N)p(1-p)^C] \quad (3.15)$$

#### **d - Cálculo da probabilidade de perda $P_{drop}$**

Calcula-se a probabilidade de perda  $P_{drop}$  para:

- sistemas com detectores lentos

$$P_{\text{drop}}(v) = \frac{v^D \gamma_f (1 - \gamma_f)^{B-1}}{1 - v^N} \times \left\{ 1 - \frac{\gamma_f [1 - (1 - \gamma_f) v^{2N}]}{[1 - (1 - \gamma_f) v^N]^2} \right\} + \frac{\gamma_f^2 v^N}{[1 - (1 - \gamma_f) v^N]} + \gamma_f^2 (1 - \gamma_f)^{B-1} (v^D - v^{BN}) \quad (3.16)$$

- sistemas com detectores rápidos

$$P_{\text{drop}}(v) = \frac{v^D \gamma_{mf} (1 - \gamma_{mf})^{B-1}}{1 - v^N} \times \left\{ 1 - \frac{\gamma_{mf} [1 - (1 - \gamma_{mf}) v^{2N}]}{[1 - (1 - \gamma_{mf}) v^N]^2} \right\} + \frac{\gamma_{mf}^2 v^N}{[1 - (1 - \gamma_{mf}) v^N]} + \gamma_{mf}^2 (1 - \gamma_{mf})^{B-1} (v^D - v^{BN}) \quad (3.17)$$

### III.1.1.3 - Roteiro Utilizado pelo Programa

Para o cálculo da perda de pacote, chegou-se à expressão dada por (3.18):

$$P_{\text{drop}} = \sum_{R=0}^{N-1} \sum_{C=0}^{M-R} \Theta_R(R) \Theta_{C/R}(C/R) P_{\text{drop}}(v) \quad (3.18)$$

Para calcular esta expressão da probabilidade de perda, o programa segue as etapas abaixo:

#### a - Vetor $\Theta(R)$

Como o sistema pode admitir valores de "slots" reservados desde zero até o número máximo de "slots" menos 1, que é  $(N - 1)$ , e como  $R$  é o número de "slots" reservados num determinado "frame", pode-se construir o vetor  $\Theta_R(R)$ , de dimensão  $N$ , como sendo:

$$\Theta_R(R) = [\theta(0), \theta(1), \theta(2) \dots \theta(N-1)] \quad (3.19)$$

#### b - Matriz $\Theta(C/R)$

Com a expressão (3.13), encontra-se  $\Theta_{C/R}(C/R)$ , com  $c$  calculado anteriormente e com  $p_o$  fixo no passo.

**TetR** -> Nome da sub-rotina

$$\theta_{C/R}(C/R) = \begin{cases} p_o(1-p_o)^C, C < M-R \\ (1-p_o)^{M-R}, C = M-R \\ 0, \text{outros valores} \end{cases} \quad (3.20)$$

Obs.: deve-se observar a diferença entre  $C$  e  $c$ ;  $C$  é a quantidade de terminais em disputa que pode variar desde zero até todos os  $N$  possíveis terminais e  $c$  é o número de terminais em disputa no equilíbrio.

Os limites de  $R$  já foram definidos e os de  $C$  são 0 e  $M$ , pois o valor máximo de terminais em disputa num sistema com  $M$  possíveis terminais é justamente  $M$ . Assim obtém-se a matriz  $\Theta(C/R)$  de dimensões  $N \times (M+1)$  na forma:

$$\theta_{C/R}(C/R) = \begin{bmatrix} \theta(0/0) & \theta(0/1) & . & . & . & . & . & . & \theta(0/N-1) \\ \theta(1/0) & \theta(1/1) & . & . & . & . & . & . & \theta(1/N-1) \\ \theta(2/0) & \theta(2/1) & . & . & . & . & . & . & \theta(2/N-1) \\ . & . & . & . & . & . & . & . & . \\ \theta(M-R/0) & \theta(M-R/1) & . & . & . & . & . & . & \theta(M-R/N-1) \\ \theta(M-R+1/0) & \theta(M-R+1/1) & . & . & . & . & . & . & 0 \\ . & . & . & . & . & . & . & 0 & 0 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ \theta(M/0) & 0 & . & . & . & . & . & . & . \end{bmatrix} \quad (3.21)$$

### **c - Matriz $V(C,R)$**

Neste passo calcula-se  $V(C,R)$  que é também uma matriz de dimensões  $N \times (M+1)$ .

onde:



$$v(C,R) = Ve(C,R) = 1 - (1 - \gamma_m) \left(1 - \frac{R}{N}\right) p(1-p)^C \quad (3.22)$$

$v(C,R)$  - Notação utilizada no texto.

$Ve(C,R)$  - Notação utilizada nas subrotinas cujo nome é **Ve**.

$Ve(C,R) =$

$$= \begin{bmatrix} Ve(0,0) & Ve(0,1) & . & . & . & . & . & . & Ve(0,N-1) \\ Ve(1,0) & Ve(1,1) & . & . & . & . & . & . & Ve(1,N-1) \\ Ve(2,0) & Ve(2,1) & . & . & . & . & . & . & Ve(2,N-1) \\ . & . & . & . & . & . & . & . & . \\ Ve(M-R,0) & Ve(M-R,1) & . & . & . & . & . & . & Ve(M-R,N-1) \\ Ve(M-R+1,0) & Ve(M-R+1,1) & . & . & . & . & . & . & Ve(M-R+1,N-1) \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ Ve(M,0) & Ve(M,1) & . & . & . & . & . & . & Ve(M,N-1) \end{bmatrix} \quad (3.23)$$

**d - P<sub>drop</sub>(v)**

Calcula-se **P<sub>drop</sub>[Ve(C,R)]** que nada mais é do que um mapeamento de **Ve** e desta forma tem a mesma dimensão de **Ve(C,R)**. Esta matriz é encontrada pela expressão (3.03) ou (3.04) dependendo do modelo adotado (detectores de atividade lentos ou rápidos, respectivamente).

**Pd(C,R)** - Notação utilizada no texto.

**pdv** - Nome da subrotina.

$Pd [Ve (C, R ) ] = pdv =$

$$= \begin{bmatrix} \text{pdv}(0,0) & \text{pdv}(0,1) & . & . & . & . & . & \text{pdv}(0,N-1) \\ \text{pdv}(1,0) & \text{pdv}(1,1) & . & . & . & . & . & \text{pdv}(1,N-1) \\ \text{pdv}(2,0) & \text{pdv}(2,1) & . & . & . & . & . & \text{pdv}(2,N-1) \\ . & . & . & . & . & . & . & . \\ \text{pdv}(M-R,0) & \text{pdv}(M-R,1) & . & . & . & . & . & \text{pdv}(M-R,N-1) \\ \text{pdv}(M-R+1,0) & \text{pdv}(M-R+1,1) & . & . & . & . & . & \text{pdv}(M-R+1,N-1) \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ \text{pdv}(M,0) & \text{pdv}(M,1) & . & . & . & . & . & \text{pdv}(M,N-1) \end{bmatrix} \quad (3.24)$$

Como a probabilidade de perda de pacote é dada por (3.18), pode-se escrever esta equação na forma:

$$P_{\text{drop}} = \sum_{R=0}^{N-1} \Theta_R (R) \sum_{C=0}^{M-R} \text{pd1}(C, R) \quad (3.25)$$

onde  $\text{pd1}(C/R)$  tem a forma:

$$\text{pd1}(C, R) = \Theta_{C/R} (C/R) \circ P_{\text{drop}} (v) \quad (3.26)$$

e  $[\circ]$  indica o produto elemento a elemento das matrizes (isto indica que as duas matrizes deverão ter as mesmas dimensões, assim como a matriz resultante).

$$P_{\text{drop}} (C, R) =$$

$$= \begin{bmatrix} \text{Pd1}(0/0) & \text{Pd1}(0/1) & . & . & . & . & . & \text{Pd1}(0/N-1) \\ \text{Pd1}(1/0) & \text{Pd1}(1/1) & . & . & . & . & . & \text{Pd1}(1/N-1) \\ \text{Pd1}(2/0) & \text{Pd1}(2/1) & . & . & . & . & . & \text{Pd1}(2/N-1) \\ . & . & . & . & . & . & . & . \\ \text{Pd1}(M-R/0) & \text{Pd1}(M-R/1) & . & . & . & . & . & \text{Pd1}(M-R/N-1) \\ \text{Pd1}(M-R+1/0) & \text{Pd1}(M-R+1/1) & . & . & . & . & . & 0 \\ . & . & . & . & . & . & 0 & 0 \\ . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . \\ \text{Pd1}(M/0) & 0 & . & . & . & . & . & 0 \end{bmatrix} \quad (3.27)$$

Cria-se um vetor  $\mathbf{x}$  de dimensão  $1 \times (M+1)$  com o objetivo de calcular a expressão  $\sum_{C=0}^{M-R} \text{pd1}(C, R)$

$$\mathbf{x} = [111 \dots 11] \quad (3.28)$$

Multiplicando o vetor  $\mathbf{x}$  pela matriz **pd1** obtém-se um outro vetor, que transposto chamar-se-á de **pd2** e é de dimensão  $(1 \times N)$ .

$$\text{pd2}(R) = \mathbf{x} \text{pd1}(C, R) = \sum_{C=0}^{M-R} \text{pd1}(C, R) = \quad (3.29)$$

$$= [111 \dots 1] \times \begin{bmatrix} \text{pd1}(1,1) & \text{pd1}(1,2) & . & . & \text{pd1}(1,N) \\ \text{pd1}(2,1) & \text{pd1}(2,2) & . & . & \text{pd1}(2,N) \\ \text{pd1}(3,1) & \text{pd1}(3,2) & . & . & \text{pd1}(3,N) \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ \text{pd1}(M+1,1) & . & . & . & \text{pd1}(M+1,N) \end{bmatrix} \quad (3.30)$$

Finalmente ao multiplicar-se  $\theta_R(R)$  por **pd2**, obtém-se a probabilidade de perda de pacotes **P<sub>drop</sub>** onde:

$$P_{\text{drop}} = \sum_{R=0}^{N-1} \Theta_R(R) \text{pd2}(R) = \Theta_R(R) \text{pd2}(R) \quad (3.31)$$

#### III.1.1.4 - Resultados Obtidos para Sistemas com Voz Apenas

Expõe-se a seguir os resultados obtidos no cálculo de **P<sub>drop</sub>** para vários valores de **p**. Os resultados para os quais o sistema encontra-se em uma região instável, foram descartados.

### Detectores de atividade de voz lento

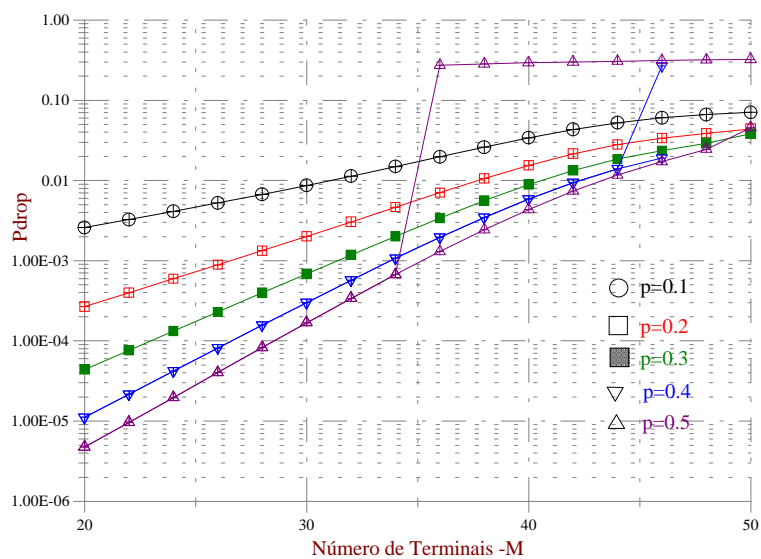


fig. 3.03

### Detectores de voz rápidos

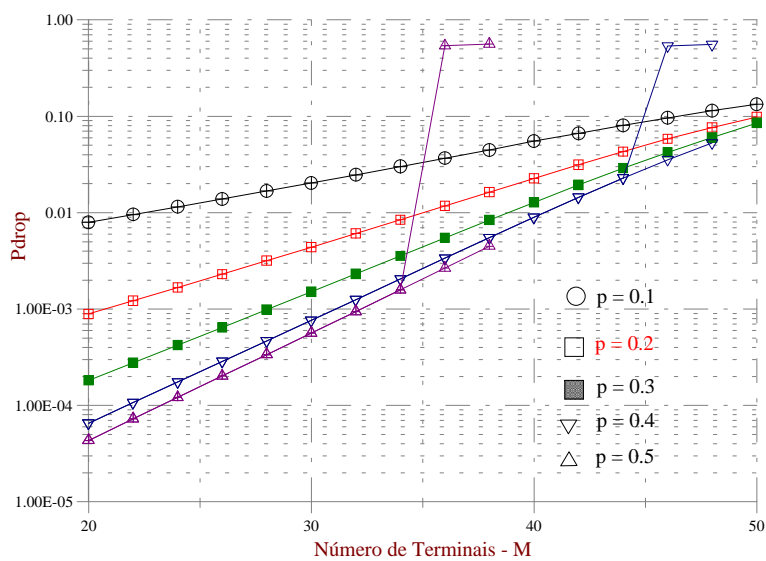


fig. 3.04

Observa-se destas figuras que fixando-se a probabilidade de perda de pacotes e a de acesso, o número de possíveis conversações simultâneas varia, conforme o detector de atividade vocal considerado. Mas esta variação tende a diminuir à medida que se aumenta  $p$ , dentro do intervalo de valores ensaiados. Por outro lado, aumentar  $p$  significa que os terminais competirão com mais frequência, decrescendo o tempo de espera e a probabilidade de perda de pacotes.

### III.1.1.5 - Comparação entre os Vários Métodos para uma dada Probabilidade de Acesso

As fig. 3.05 a 3.08 mostram os resultados obtidos para a probabilidade de perda de pacotes calculada por métodos distintos, utilizando uma mesma probabilidade de acesso  $p$ .

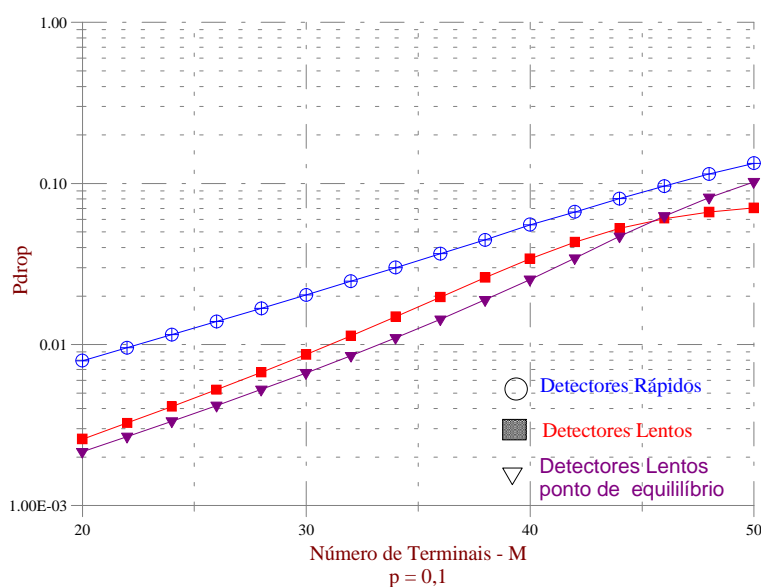


fig. 3.05

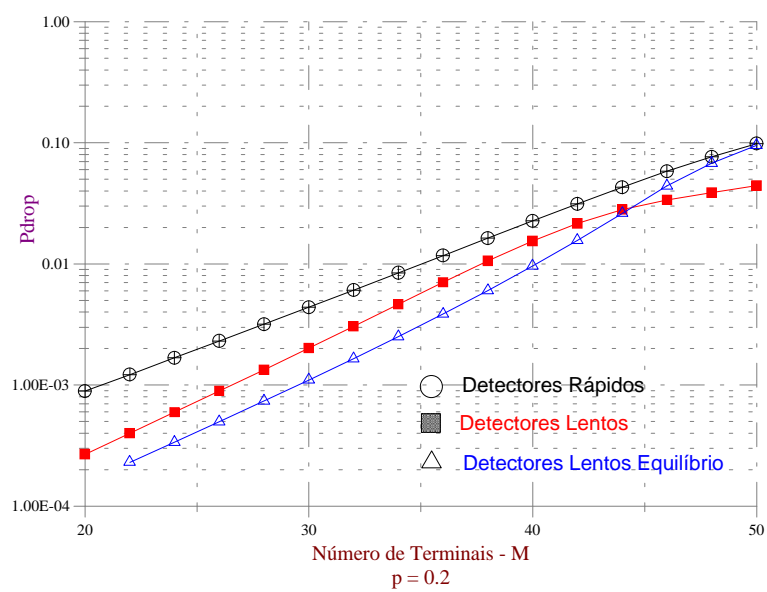


fig. 3.06

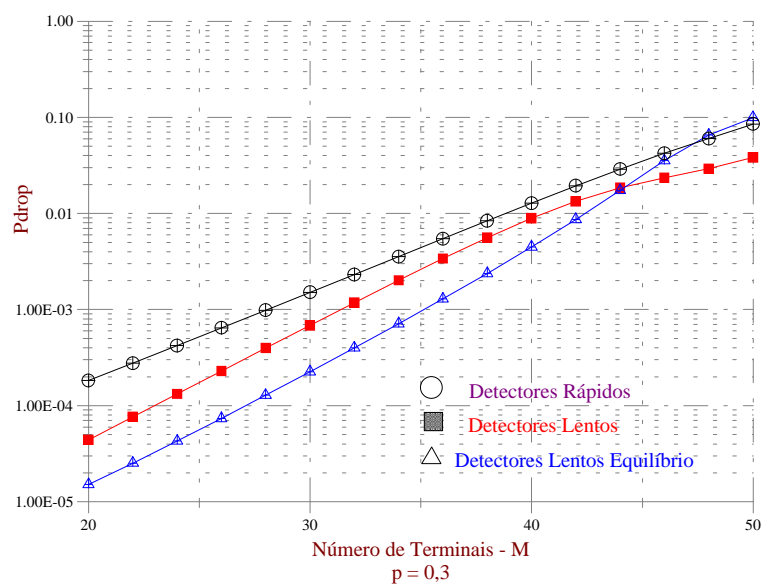


fig. 3.07

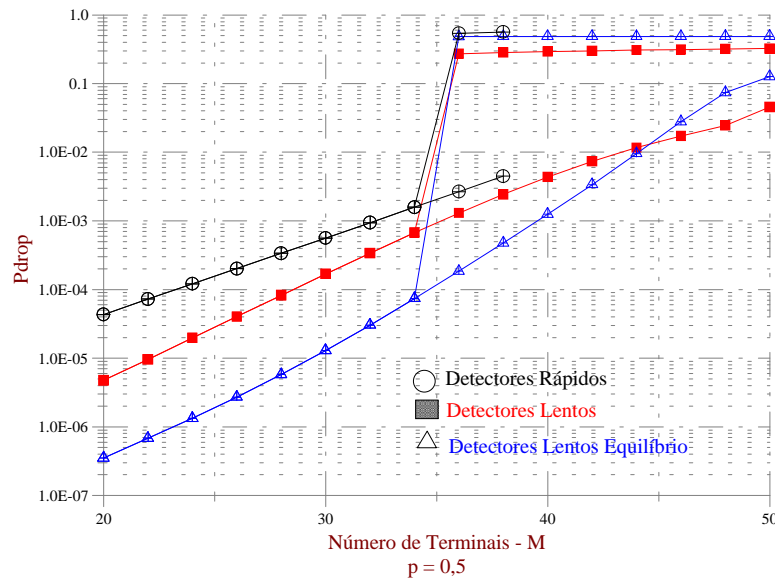


fig. 3.08

A simplicidade aliada ao processo de cálculo pelo ponto de equilíbrio e seus resultados próximos à medida que se aumenta  $p$ , levaram ao seu uso quando da introdução de dados. Desta forma, a partir deste ponto abandonam-se os demais métodos de cálculo e os seguintes serão baseados exclusivamente em detectores lentos e ponto de equilíbrio.

#### III.1.1.6 - Influência da Duração do "Frame"

Neste item verifica-se a influência da duração do "frame" na probabilidade de perda de pacotes  $P_{drop}$ , pois quando  $T$  é pequeno, o cabeçalho fixo reduz a eficiência do sistema. Com valores maiores de  $T$ , cada colisão prejudica mais o canal e por mais tempo.

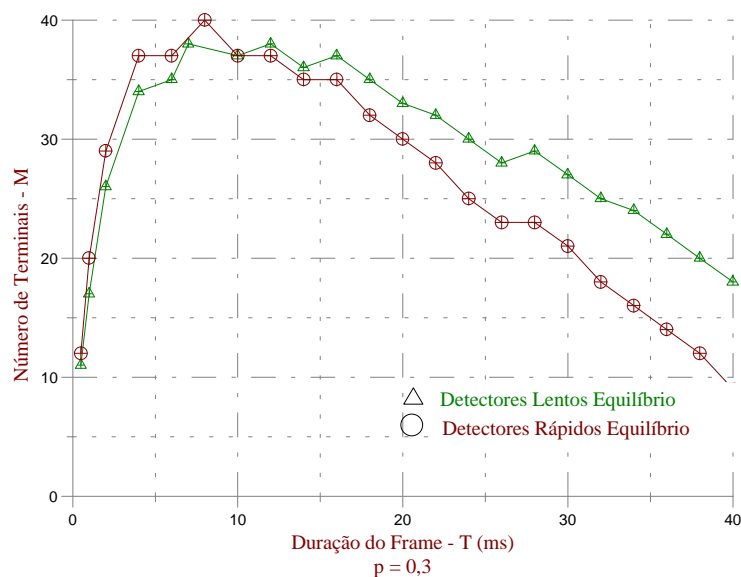


fig. 3.09

Para os resultados apresentados na fig. 3.09 fixa-se uma probabilidade de acesso  $p=0.3$  e um valor para  $D_{\text{máx}}$  igual a 32ms. Variando-se a duração do "frame"  $T$  e fixando-se a demais variáveis, calcula-se a probabilidade de perda de pacotes e encontra-se o número de terminais suportados para uma perda de 1%.

Observa-se na figura que o detector rápido tem um melhor desempenho que o detector lento para "frames" menores, mas quando a duração do "frame" aumenta, o desempenho dos detectores lentos supera o desempenho dos detectores rápidos devido ao fato que para conseguir uma reserva, primeiro ele deverá observar o "slot" livre e no próximo "slot" tentar a disputa. Como o tamanho do "frame" é longo, o próximo "frame" daquele "slot" provavelmente conterà voz provocando assim uma degradação no sistema.



## III.2 - SISTEMAS PRMA COM DADOS ACOPLADOS

Neste item evidenciam-se os resultados obtidos para sistemas com detectores de atividade vocal lento, vistos no capítulo 2 em seu item III.2. No cálculo da  $P_{drop}$  para sistemas com dados acoplados pode-se seguir o método proposto no item II.4.3.2, baseado no ponto de equilíbrio.

### III.2.1 - Resultados Obtidos

A seguir tem-se os resultados obtidos com o método II.4.3.2 para vários valores da probabilidade de acesso de terminais de voz  $p$  e da probabilidade de acesso dos terminais de dados  $pd$ . Observe-se em primeiro lugar que das soluções obtidas representam-se apenas as estáveis. Na figura 3.10, a seguir, tem-se um exemplo de ilustração para o caso instável onde observa-se que numa das soluções, à medida que se aumentam o número de terminais a probabilidade de perda diminui, em nítida discordância com o resultado físico esperado.

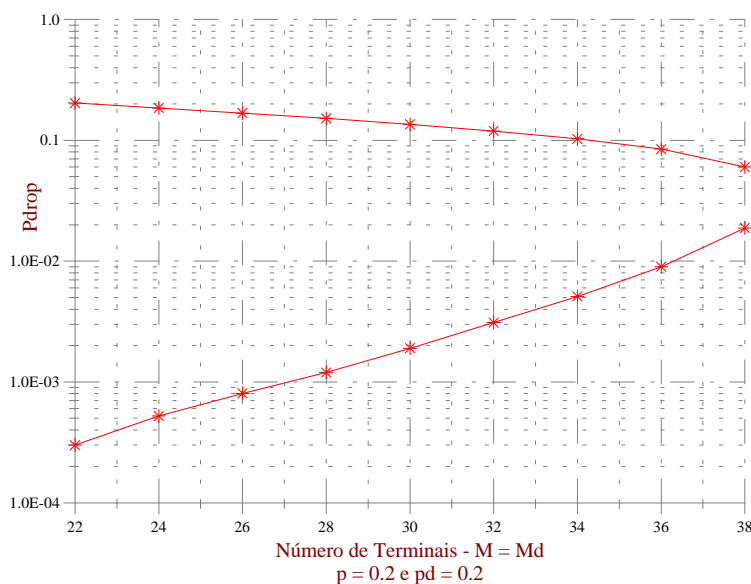


fig 3.10

A figura representa os resultados obtidos para um sistema com dados acoplados admitindo-se  $p=0.2$  e  $pd=0.2$ , com detectores de atividades vocais lentos e calculados pelo ponto de equilíbrio. Como mencionado, observa-se que o sistema admite duas soluções distintas de probabilidade de perda de

pacotes de dados, para um determinado número de terminais **M**. Observando-se o sistema adotado na fig. 2.18 e de acordo com o item II.3.10, pode-se equacionar o fluxo de saída do estado de disputa deste sistema como sendo a função  $G(c)$  dada pela equação 3.32.

$$G(c) = \text{cpu}(1 - \gamma)(1 - r) + c\gamma \quad (3.32)$$

Baseando-se no item II.3.10, pode-se admitir que o sistema encontra-se em um ponto de equilíbrio estável, quando a derivada de  $G(c)$  em relação a  $c$  for positiva e instável caso contrário. Esta derivada pode ser calculada pela equação:

$$\frac{dG(c)}{dc} = \frac{\partial G(c)}{\partial c} + \frac{\partial G(c)}{\partial b} \frac{db}{dc} \cong \frac{\partial G(c)}{\partial c} + \frac{\partial G(c)}{\partial b} \frac{\Delta b}{\Delta c} \quad (3.33)$$

Efetuada-se o cálculo da derivada, através de (3.33), da expressão (3.32), nota-se que nestes pontos de solução a derivada é sempre negativa e assim estes pontos não são considerados nos resultados adiante.

Nas duas figuras a seguir (3.11 e 3.12) ilustra-se o efeito da variação da probabilidade de perda de pacotes em função do número de terminais, tendo como parâmetro a probabilidade de acesso dos terminais de dados **pd**, para um valor fixo da probabilidade de acesso dos terminais aos canais vocais **p**.

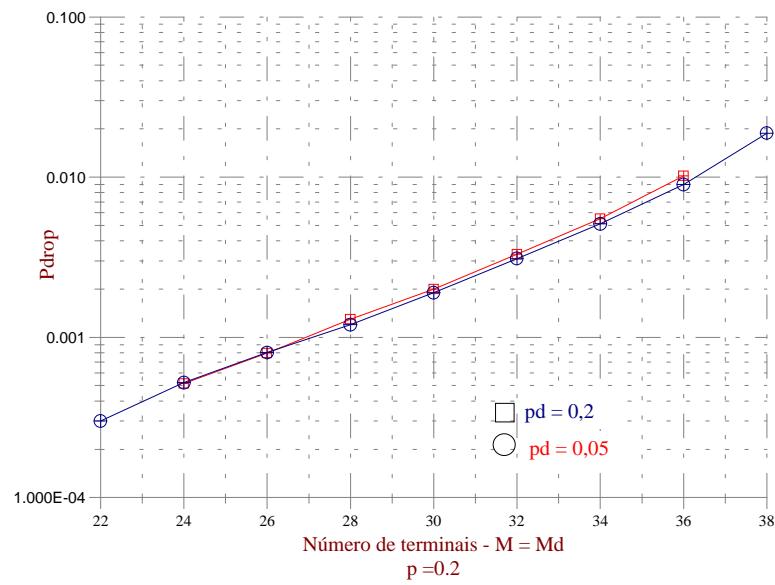


fig. 3.11

Modelo com detector lento de voz e com dados acoplados para  $p=0.2$  e dois valores de  $pd$ .

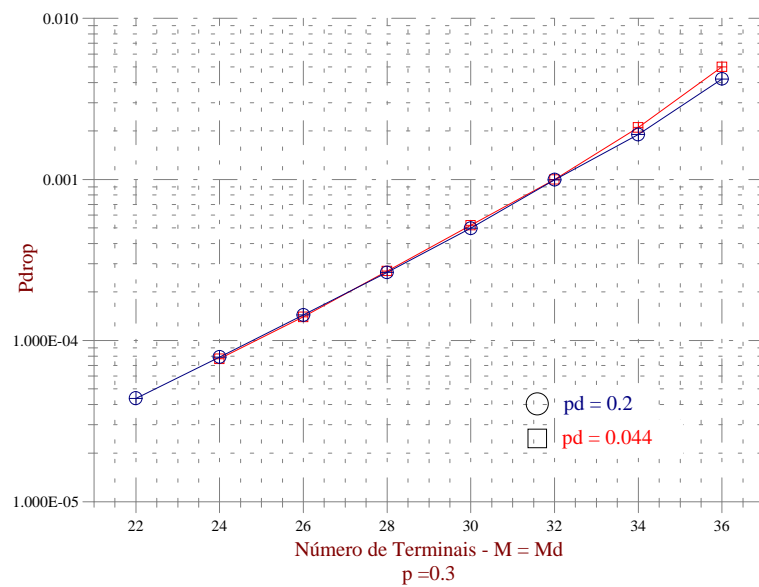


fig. 3.12

Modelo com detector lento de voz e com dados acoplados para  $p=0.3$  e dois valores de  $pd$ .

Observe-se na fig. 3.11 que para uma probabilidade de perda de pacotes de 0,01 pode-se ter 36 conversações simultâneas com a transmissão de dados à taxa de 1200 bps para todos os terminais, enquanto sem dados este número é 40 (conforme pode se verificar na fig.3.06).

Note-se ainda que a probabilidade de perda de pacotes de voz é praticamente independente da probabilidade de acesso dos dados, e isso é devido a um aumento na fila destes dados nos terminais, já que estes não tem reserva para o envio. Este aumento na fila provoca atrasos no envio e as fig. 3.13 e fig. 3.14 evidenciam estes atrasos médios (em milissegundos) calculados pelas equações (2.57) e (2.60), para um sistema com detectores lentos de atividade de voz e probabilidade de permissão iguais  $p=0.3$  e  $p=0.2$ , respectivamente.

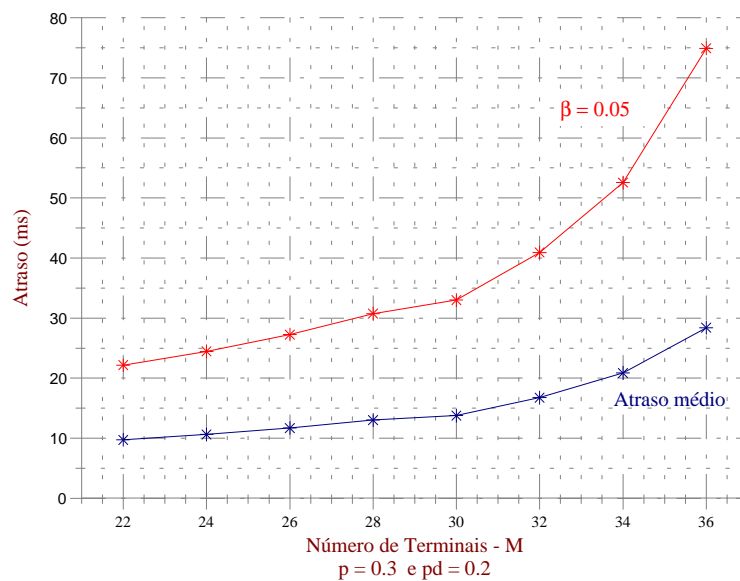


fig. 3.13

Desta figura considerando-se o ponto de funcionamento anterior (36 conversações simultâneas com transmissão de dados a 1200 bps para todos os terminais) verifica-se que o atraso médio na transmissão dos pacotes com dados é de  $\bar{\omega}=28$  ms e  $P(\omega \geq 75 \text{ ms}) \leq 0,05$ .

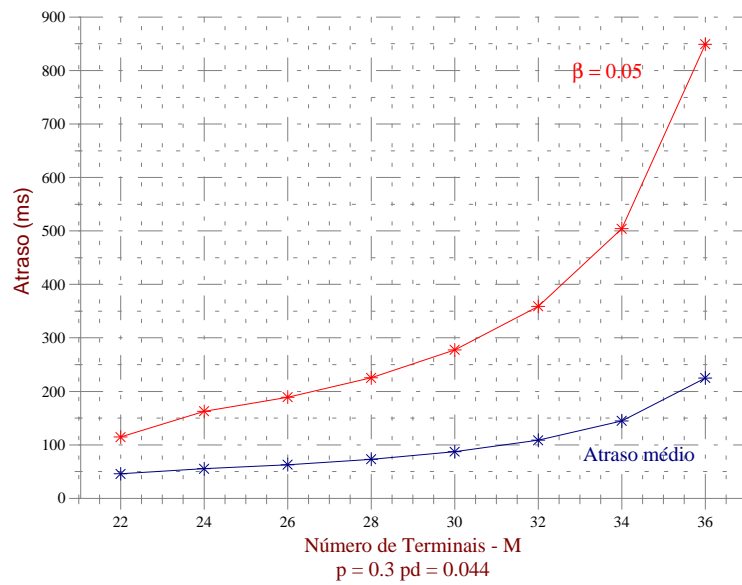


fig. 3.14

Nas figuras 3.15 e 3.16 tem-se os resultados da probabilidade de perda de pacotes para um valor constante da probabilidade de acesso de dados **pd** e dois valores distintos da probabilidade de acesso dos terminais de voz **p**.

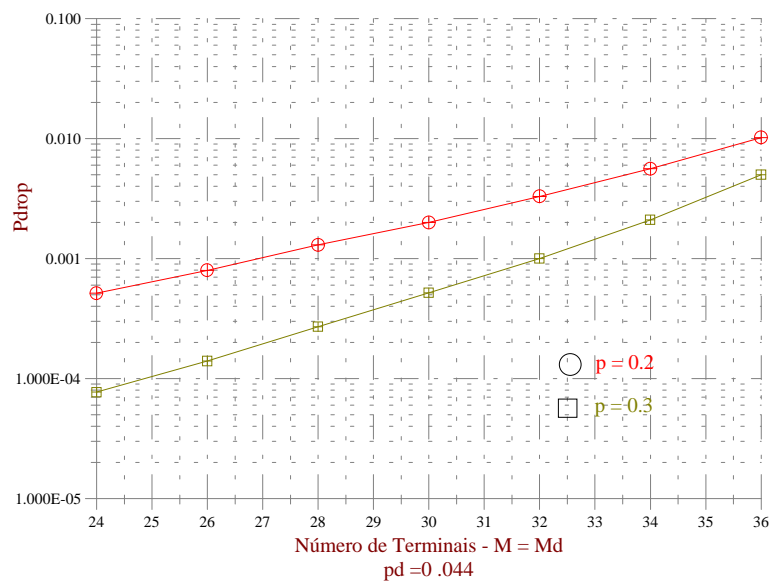


fig. 3.15

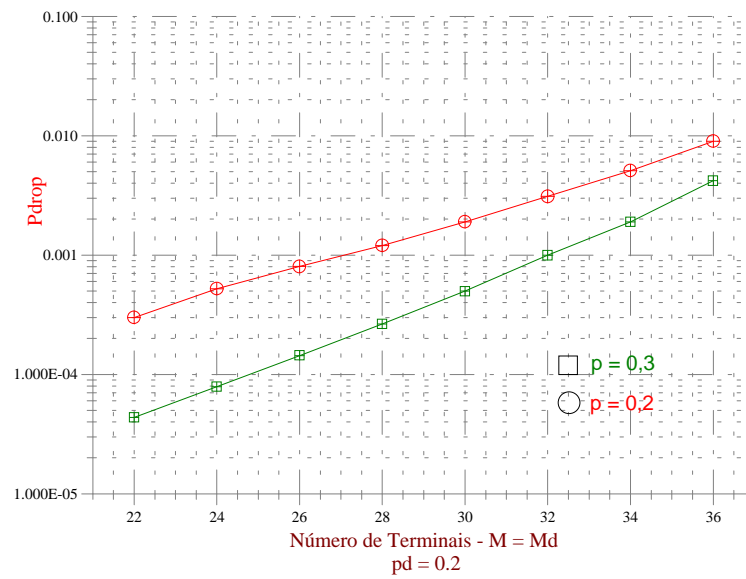


fig. 3.16

As figuras evidenciam ganhos de capacidade à medida que aumenta-se  $p$ , como já observado anteriormente, e uma degradação devida aos tráfego simultâneo de voz e dados.

Na fig. 3.17 comparam-se os resultados obtidos com o sistema com apenas voz e o sistema com voz e dados acoplados.

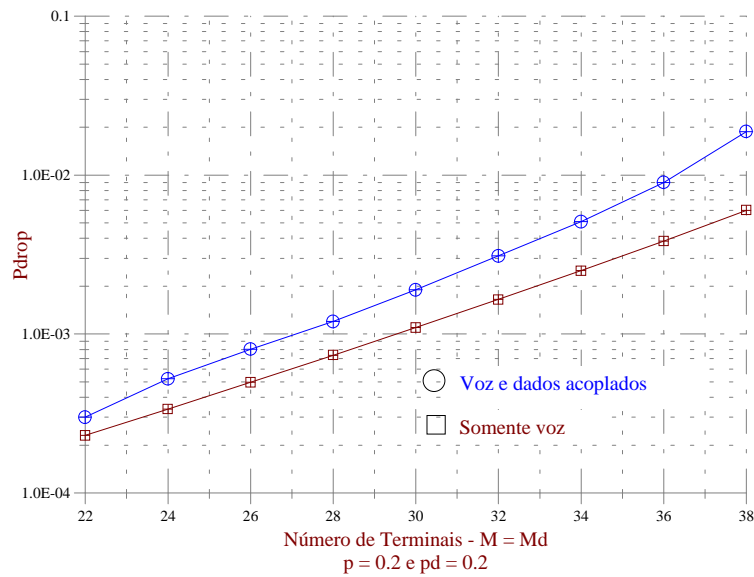


fig. 3.17

Observa-se que a introdução de dados, com um valor pré-estabelecido de  $M=M_d$ , há uma degradação na probabilidade de perda de pacotes, como já notado anteriormente.

Na figura 3.18 ilustra-se, uma comparação entre os sistemas com dados acoplados mas com o número de terminais de dados fixo e pequeno e para o valor de  $M$  variando. Esta figura também permite verificar que os resultados obtidos para os valores da probabilidade de perda de pacotes de voz em um sistema com dados acoplados tem como limite inferior a probabilidade de perda de pacotes de voz calculado para sistemas sem terminais de dados. Nesta figura utilizou-se uma probabilidade de acesso de voz  $p = 0,2$  e uma probabilidade de acesso de dados  $p_d = 0.1$

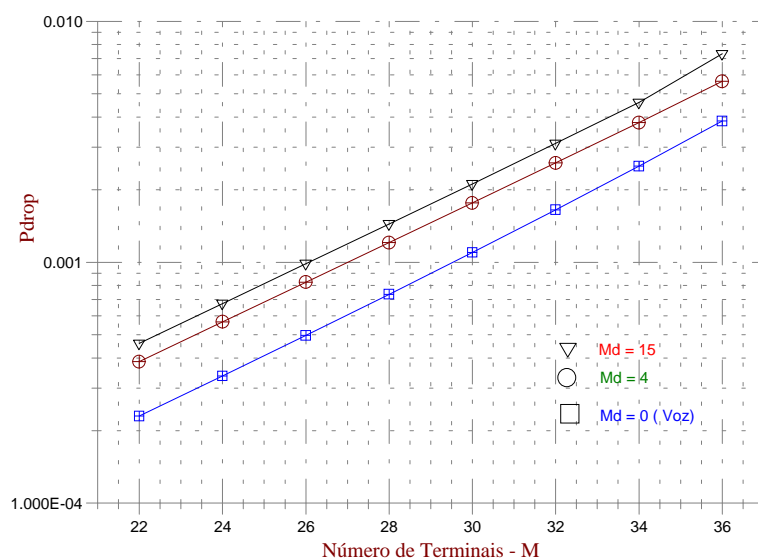


fig. 3.18

Nota-se por esta figura, por exemplo, que a inclusão de 4 terminais de dados à taxa de 1200 bps, com uma probabilidade de perda de pacotes de 0,01 reduz o número de conversações simultâneas de 29 para 27, apenas.

### III.2.2 Exemplo de Sistema PRMA

Descreve-se neste item um sistema PRMA com duas taxas de transmissão distintas: de 16 Kbps e 32 Kbps. Na tabela a seguir apresentam-se algumas características destes dois sistemas calculados segundo metodologia apresentada nos itens anteriores. Na figura 3.19 ilustra-se o desempenho destes dois sistemas e o valor considerado para a determinação do número de conversações simultâneas é  $P_{drop} = 1\%$



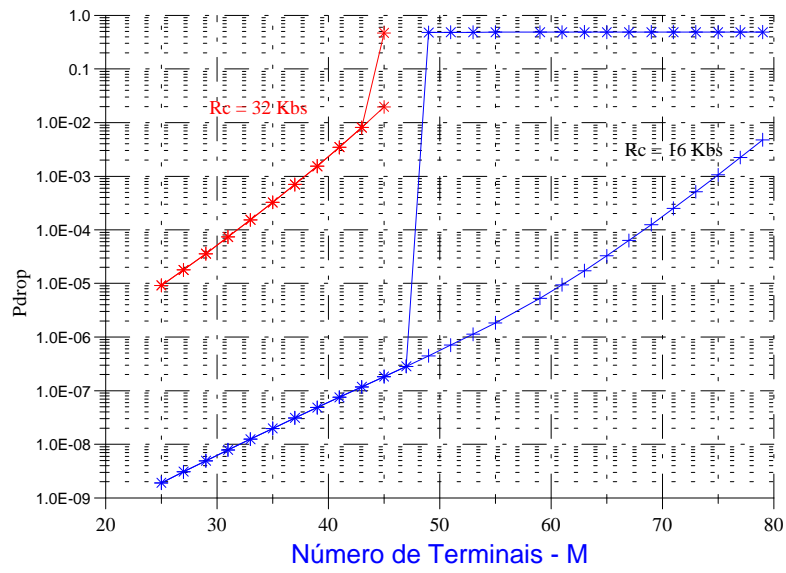


fig. 3.19

Taxa da Fonte (Kbps)	16	32	$R_s$
Tamanho do Pacote (bits)	576	576	$R_s T + H$
Bits de voz (bits)	512	512	$R_s T$
Bits de cabeçalho (bits)	64	64	$H$
Taxa de Transmissão do canal (Kbps)	720	720	$R_c$
Duração do 'Frame' (ms)	32	16	$T$
Slots por frame	40	20	$N = \lfloor R_c T / (R_s T + H) \rfloor$
Atraso Máximo para voz (ms)	32	32	$D_{\max}$
Duração do slot (ms)	0,8	0,8	$\tau$
Tempo máximo de espera - (slots)	1	2	$D = \lfloor D_{\max} / \tau \rfloor$
Probabilidade de permissão	0,4	0,4	$p$
Capacidade de Terminais	80	43	$M (1\%)$
Canais equivalentes	45	22,5	$R_c / R_s$
Terminais por canal	1,78	1,91	$M_{1\%} R_c / R_s$
"Throughput"	619,2	665,6	$r$
Canais PRMA por MAN	80,7	75,1	$z = 100.000 / 2r$
Mínima conversação / base	50	50	$z1 = \lfloor z / M_{1\%} \rfloor$
Canais PRMA por base	1	2	$z2 = z1 / M_{1\%}$
Estações bases por MAN	81	38	$z3 = z / z2$
Erlangs por Estações base	54,4	49,6	$z4$
Erlangs por MAN	4406,4	1884,8	$z5 = z3 / z4$
Distância entre estações bases (m)	50	50	$z6$
Largura da grade das estações (m)	10	7	$z7$
Comp. da grade das estações (m)	9	7	$z8$
Array entre estações bases (m)	90	49	$z9 = z7 * z8 < z6$
Área de serviço (Km2)	0,225	0,1225	$z10 = (z6)^2 z9 * E - 6$
Fator de reuso celular	25	25	$z11$
Banda Tx e Rx - MHz	2	4	$z12$
Banda Total - MHz	50	100	$z13 = z11 * z12$
Erlang/Km2/MHz	391,6	153,8	$z14 = z5 / (z13 * z10)$

onde:

$$r = \frac{R_s M_{1\%}}{N} \frac{t_1}{t_1 + t_3}$$

$M(1\%)$  : número de terminais que o sistema suporta com uma probabilidade de perda de pacotes de 0,01.

Observando os resultados acima obtidos, verifica-se que dentro das características aqui adotadas este sistema supera outros atualmente em uso. O resultado da última linha indica que o sistema tem uma capacidade para escoar 391 Er/Km<sup>2</sup>/MHz, com taxa de transmissão de 16 Kbps..

### **III.3 - CONCLUSÕES**

Neste trabalho procurou-se desenvolver uma metodologia para o cálculo de desempenho de sistemas PRMA com pacotes de voz e dados acoplados. Os resultados evidenciam sua possível aplicação em sistemas microcelulares com resultados promissores.

O envio compartilhado de dados mostra-se factível, com uma pequena degradação tolerável para a aplicação requerida, principalmente para baixos valores  $M_d$ , como é de se esperar numa aplicação real.

Este trabalho é no entanto introdutório à questão, pois várias pendências precisam ser ainda melhor estudadas, como por exemplo: robustez quanto a erros de transmissão; consequências devidas às filas finitas para os dados, efeito de captura, possibilidade do uso de algoritmos para alocação dinâmica de canais, estudos do congestionamento de sistemas com dados acoplados que obtêm reserva de "slots" etc.

Refinamentos do estudo aqui apresentado, assim como investigações para a solução das pendências apontadas, serão objeto de um futuro trabalho a ser elaborado.

## **IV - REFERÊNCIAS BIBLIOGRÁFICAS**

- [1] D. J. Goodman, "**Cellular Packet Communications**", IEEE Transactions on Communications, Vol. 38, No. 8, pp.1272-1280, August 1990.
- [2] S. Nanda, D. J. Goodman e U. Timor, "**Performance of PRMA: A Packet Voice Protocol for Cellular Systems**", IEEE Transactions on Vehicular Technology, Vol. 40, No. 3, pp. 384-398, August 1991.
- [3] D. J. Goodman, R. A. Valenzuela, K. T. Gayliard e B. Ramamurthi, "**Packet Reservation Multiple Access for Local Wireless Communications**", IEEE Transactions on Communications, Vol. 37, No. 8, pp. 885-890, August 1989.
- [4] V. O. K. Li, "**Multiple Access Communications Networks**", IEEE Communications Magazine, Vol. 25, No. 6, pp.41-48, June 1987.
- [5] C. E. Sundberg "**Continuous Phase Modulation**" IEEE Communications Magazine, Vol. 24, No. 4, pp. 25-38, April 1986.
- [6] F. A. Tobagi, "**Multiaccess Protocols in Packet Communication Systems**", IEEE Transactions on Communications, Vol. 28, No. 4, pp. 468-488, April 1980.
- [7] A. M. Saleh e L. J. Cimini Jr., "**Indoor Radio Communications using Time-Division Multiple Access with Cyclical Slow Frequency Hopping and Coding**", IEEE Journal on Selected Areas in Communications, Vol. 7, pp.59-70, January 1989.
- [8] L. F. Chang e J. C. Chuang, "**Diversity Selection using Coding in a Portable Radio Communications Channel with Frequency-Selective Fading**", IEEE Journal on Selected Areas in Communications, Vol. 7, pp.89-98, January 1989.
- [9] R. D'Avella, L. Moreno e M. Sant'Agostino, "**An Adaptive MLSE Receiver for TDMA Digital Mobile Radio**", IEEE Journal on Selected Areas in Communications, Vol. 7, pp.122-129, January 1989.

- [10] T. A. Sexton e K. Pahlavan, "**Channel Modeling and Adaptive Equalization in Indoor Radio Channels**", IEEE Journal on Selected Areas in Communications, Vol. 7, pp.114-121, January 1989.
- [11] L. G. Roberts, "**ALOHA Packet System with and without Slots and Capture**", ComputerCommunications Revue, Vol. 5, pp. 28-42, April 1975.
- [12] S. Nanda, "**Analysis of Packet Reservation Multiple Access: Voice and Data Integration for Wireless Networks**", in Proc. IEEE GLOBECOM '90, San Diego, CA, pp. 1984-1988, Dec. 1990.
- [13] S. Tasaka, "**Stability and Performance of the R-ALOHA Packet Broadcast System**", IEEE Transactions on Computers, Vol. C-32, No. 8, pp. 717-726, August 1983.

## **ANEXO 1**

### **LISTAGEM COMENTADA DAS SUBROTINAS UTILIZADAS.**

Neste anexo são descritas as várias subrotinas desenvolvidas ao longo deste trabalho, designadas sequencialmente por A1.i. Esta descrição segue um padrão na forma:

**Nome da subrotina**  
**Descrição**  
**Variáveis de entrada**  
**Variáveis de saída**  
**Listagem comentada**

#### **A1.1**

**% Nome da subrotina: INI**

**% Descrição:**

% Esta subrotina inicia o aplicativo, colhendo dados do usuário afim de efetuar uma  
 % das várias possibilidades de cálculo disponíveis.

**% Variáveis de entrada:**

% TIPER Esta variável pode assumir os valores:  
 % 1 - Quando o usuário desejar calcular a probabilidade de perda de pacotes.  
 % 2 - Quando o usuário desejar calcular o número de terminais que o sistema suporta para  
 % uma Pdrop=0.01.  
 % MODELO: Esta variável pode assumir os valores:  
 % 1 - Sistema com detectores de voz rápidos.  
 % 2 - Sistema com detectores de voz lentos calculado utilizando dados do ponto de  
 % equilíbrio.  
 % 3 - Sistema com detectores de voz lentos.  
 % 4 - Sistema com dados acoplados.  
 % 5 - Sistema com detectores de voz rápidos, calculado utilizando  
 % os dados do ponto de equilíbrio.

**% Variáveis de saída:**

% Esta subrotina não tem variáveis de saída.

**% Listagem comentada:**

% Preparando a tela inicial

clear all

clear, format, figure, clc, hold off

for ccc=1:30, close, end

% Entrada das variáveis TIPER e MODELO.

tiper=menu('Tipo de cálculo de performance desejada', ...

'[probabilidade de Perda de pacotes]',...

'[Performance da duração do frame - T]');

% O programa pode calcular a probabilidade de Perda de pacotes (Pdrop) em função do

% número de terminais em disputa ou pode calcular o número de terminais (M) que o sistema

% suporta para Pdrop=0.01; fornecer este número à subrotina INI0.m afim de construir a

% curva  $M=f(T)$ .

if tiper=1

modelo=menu('Escolha o tipo de sistema que deseja calcular a Pdrop', ...

```

        '[Sistema com detectores de voz rápidos]',...
        '[Sistema com detectores de voz lentos - [pto equilíbrio]]',...
        '[Sistema com detectores de voz lentos ]', ...
        '[Sistema com dados acoplados]');
else
modelo=menu('Escolha o tipo de sistema que deseja calcular a Pdrop', ...
        '[Sistema com detectores de voz rápidos]',...
        '[Sistema com detectores de voz lentos - [pto equilíbrio]]',...
        '[Sistema com detectores de voz lentos ]', ...
        '[Sistema com detectores de voz rápidos - [pto equilíbrio]]');
end
ini1

```

## A1.2

**% Nome da subrotina: INI1**

**% Descrição:**

Esta subrotina é chamada pela subrotina INI e tem a função de coordenar os valores mínimos e máximos de M através de contadores criados internamente.

**% Variáveis de entrada:**

```

%      P      :Probabilidade de acesso dos terminais vocais.
%      PD      :Probabilidade de acesso dos terminais de dados.
%      Mmin     :Valor inicial de M quando deseja-se encontrar a curva Pdrop=f(M).
%      Mmax     :Valor final de M quando deseja-se encontrar a curva Pdrop=f(M).
%      Tmin     :Valor inicial de T quando deseja-se encontrar a curva M=f(T).
%      Tmax     :Valor final de T quando deseja-se encontrar a curva M=f(T).
%      passo    :Valor do incremento de M ou T dependendo do tipo de cálculo a ser realizado.
%      Md       :Número de terminais de dados.

```

**% Variáveis de saída:**

```

%      cont      :Variável que assume valor inicial Mmin ou Tmin e valor final Mmax
%                ou Tmáx; tem como incremento o passo.
%      vetM(cont) :Valor da componente do vetor vetM, indexado ao valor de cont.
%      vetb(cont) :Valor da componente do vetor vetb, indexado ao valor de cont.
%      vetc(cont) :Valor da componente do vetor vetc, indexado ao valor de cont.
%      vetP(cont) :Vetor com os valores de Pdrop calculados.

```

**% Listagem comentada:**

```

% Adota-se o valor de constmod=0, detalhes ver a subrotina modif.m
constmod=0;
% Chama-se a subrotina TESTPROB.m para entrar com o valor da probabilidade de permissão
testprob
% A variável multiponto serve para evitar-se entrar várias vezes na subrotina apresentação
% quando o usuário solicita levantar a curva entre dois pontos Mmin e Mmáx ou Tmin e Tmax.
multiponto=1;
% A variável multiponto1 serve para evitar entrar-se várias vezes na subrotina apresentação
% quando o usuário solicita levantar a curva entre dois pontos Mmin e Mmáx com Md fixo.
multiponto1=1;
% Tiper=1 -> Calcula-se Pdrop=F(M)
% Tiper=2 -> Calcula-se M=f(T)
if tiper=1
    tespon
else
    entdurt

```

```

end
if Op=1
    if tiper=1
        if modelo=4
            %          Cria-se uma condicional com o a finalidade de definir se o cálculo será
            %          efetuado com Md fixo para todos os M's ou Md variando e igual a M.
            if multiponto1=1
                especMd=menu('Escolha de Md ', ...
                    '[fixo]',...
                    '[Md=M]');
                multiponto1=2;
                if especMd=1
                    disp('Entre com o valor de Md');
                    Mpj=input('Mpj=');
                end
            end
            end
            valini
        else
            dados01
        end
        aprepc
    else
        dados01
        aprepct
    end
elseif Op=2
    if tiper=1
        cont=1;
        % Cria-se este contador CONT afim de indexar-se os vetores
        % soluções b e c.
        for M=Mmin:Passo:Mmax
            vetM(cont)=M;
            % Calcula-se Pdrop para cada M desde Mmin até Mmáx
            % com incremento igual ao passo
            if modelo=4
                % Cria-se uma condicional afim de definir se o cálculo será efetuado com Md fixo
                % para todos os M's ou Md variando e igual a M.
                if multiponto1=1
                    especMd=menu('Escolha de Md ',
                        '[fixo]',
                        '[Md=M]');
                    multiponto1=2;
                    if especMd=1
                        disp('Entre com o valor de Md');
                        Mpj=input('Mpj=');
                    end
                end
            end
            valini
        else
            dados01
        end
        % Entra-se nesta subrotina afim de acertar as
        % dimensões quando o sistema contém um ponto de
        % instabilidade.
        multraiz
        cont=cont + 1;
    end
end

```



```

        % A subrotina abaixo constrói o gráfico M=f(c)
        mc
elseif tiper=2
    cont=1;
    % Cria-se este contador CONT afim de indexar o vetor soluções c.
    for T=Tmin:PassoT:Tmax
        vetT(cont)=T
        % Calcula-se Pdrop de 1% para cada T desde Mmin até
        % Mmáx com incremento igual ao passo
        M=1;
        dados01
        Pdropt2=Pdrop;
        clear Pdrop
        while Pdropt2 < .001
            Pdropt1=Pdropt2;
            dados01
            M=M + 1;
            Pdrop;
            Pdropt2=min(Pdrop);
            clear Pdrop

        end
        deltapdrop1=abs(Pdropt1 - 0.001);
        deltapdrop2=abs(Pdropt2 - 0.001);
        if deltapdrop1 > deltapdrop2
            Pdrop(cont)=Pdropt1;
        else
            Pdrop(cont)=Pdropt2;
        end
        vetM(cont)=M;
        cont=cont + 1;
    end
    % A subrotina abaixo constrói o gráfico T=f(T)
    vetT
    vetM
    mc
end
end

```

### A1.3

**% Nome da subrotina: TESTPROB**

**% Descrição:**

% Esta subrotina é chamada por INI1.m ou por MODIFIC.m. Sua principal função é testar  
 % o valor da probabilidade. Ela oferece algumas opções de probabilidade de acesso e  
 % caso o usuário deseje outro valor diferente poderá entrar com o valor que será  
 % testado para verificar se o valor digitado é coerente  $0 < p < 1$ .

**% Variáveis de entrada:**

% testp :Probabilidade de acesso dos terminais vocais.  
 % testpdad:Probabilidade de acesso dos terminais de dados.

**% Variáveis de saída:**

% p :Probabilidade de acesso dos terminais vocais.  
 % PD :Probabilidade de acesso dos terminais de dados.

**% Listagem comentada:**

% Escolha da probabilidade de acesso p

tesp=menu('Escolha o valor da probabilidade de acesso [p]', ...

'[p=0.1]','[p=0.2]','[p=0.3]',

'[p=0.4]','[p=0.5]',

'[Outro valor ?]');

% Entrada da probabilidade de acesso p, caso o usuário deseje outro valor diferente dos oferecidos.

if tesp=6

p=2;

while p > 1

disp('Entre com o valor da probabilidade de acesso p');

p=input('p=');

if p > 1

disp('Valor incorreto de probabilidade');

end

end

else

p=tesp/10

end

% A subrotina verifica se o usuário escolheu modelo com dados acoplados.

if modelo=4

if tiper=1

modelodados=1;

else

modelodados=2;

end

else

modelodados=0;

end

% Caso o usuário tenha escolhido modelo com dados acoplados o programa entra no próximo IF.

if modelodados=1

tespdad=menu('Escolha o valor da probabilidade de acesso de dados [PD]', ...

'[PD=0.044]','[PD=0.08]','[PD=0.1]','[PD=0.15]',...

'[PD=0.2]','[PD=0.25]','[PD=0.3]','[Outro valor ?]');

if tespdad=8

PD=2;

while PD > 1

disp('Entre com o valor da probabilidade de acesso de dados PD');

PD=input('Pd=');

if PD > 1

disp('Valor incorreto da probabilidade');

end

end

else

if tespdad=1

PD=0.044;

elseif tespdad=2

PD=0.08;

elseif tespdad=3

PD=0.1;

elseif tespdad=4

PD=0.15;

elseif tespdad=5

PD=0.2;

elseif tespdad=6

PD=0.25;

elseif tespdad=7

```

        PD=0.3;
    end
end
end

```

#### A1.4

##### % Nome da subrotina: TESPON

##### % Descrição:

```

%      Esta subrotina é chamada por INI1.m ou por MODIFIC.m e permite que o usuário
%      escolha se deseja efetuar o cálculo apenas para um ponto ou vários. Na segunda opção,
%      a subrotina verifica se  $M_{min} < M_{max}$ , se por qualquer motivo isto não ocorrer, o
%      aplicativo solicita ao usuário que digite novamente os valores.

```

##### % Variáveis de entrada:

```

%      Op      :Opção que pode assumir o valor 1 ou o valor 2 dependendo da opção
%               do usuário.
%      M       :Valor de M quando o usuário deseja calcular apenas para um ponto.
%      Mmin    :Valor inicial de M quando deseja-se encontrar a curva  $P_{drop}=f(M)$ .
%      Mmax    :Valor final de M quando deseja-se encontrar a curva  $P_{drop}=f(M)$ .
%      Passo   :Esta constante é o valor do incremento da variável cont (Subrot INI1).

```

##### % Variáveis de saída:

```

%      Como esta subrotina só verifica as variáveis de entrada, as de saída são as mesmas
%      da entrada.

```

##### % Listagem comentada:

```

Op=menu('Escolha a quantidade de pontos para o cálculo de  $P_{drop}$ ', ...
        '[Apenas um ponto]',...
        '[Vários pontos e curva  $P_{drop}=f(M)$ ]');
if Op==1
    disp('Digite o valor do número de terminais');
    M=input('Número de terminais=');
    M1=M;
    M=ceil(M1);
else
    Mmin=2;
    Mmax=1;
    while Mmin > Mmax
        Mmin=input('Valor inicial=');
        M1=Mmin;
        Mmin=ceil(M1);
        Mmax=input('Valor final=');
        M1=Mmax;
        Mmax=ceil(M1);
        if Mmin > Mmax
            disp('Os valores estão incoerentes')
            disp('Entre novamente com os valores')
        end
    end
    if modelo==4
        disp('Entre com o valor máximo de  $c=Mx$ ');
        Mx=input('Mx=');
        % Se o valor digitado for maior que M o programa solicita
        % para digitar novamente até ter-se digitado corretamente.
        while Mx > M

```

```

                                disp('Valor incorreto de Mx');
                                Mx=input(' Entre com o valor máximo de c=Mx ');
                                end
                                Mdx=Mx;
                                end
                                end
end
if Op=2
    disp('Digite o valor do passo ')
    Passo=input('Passo=');
end

```

### A1.5

**% Nome da subrotina: VALINI**

**% Descrição:**

```

%           Esta subrotina é chamada por INI1 e dá início ao processo de cálculo das raízes (b,c)..
%           Inicialmente calcula-se o ponto inicial A=(bo,co) e com estes valores iniciais,
%           calcula-se os valores das raízes, pelo processo de Newton Raphson.

```

**% Variáveis de entrada:**

```

%           op031 :Variável que escolhe o tipo de mapeamento por região. Pode assumir o valor 1
%           quando se deseja mapear a região inteira em sub-áreas menores. Caso contrário,
%           assume dois valores.

```

**% Variáveis de saída:**

```

%           b      :Número de terminais de dados em disputa no equilíbrio.
%           c      :Número de terminais de voz em disputa no equilíbrio.

```

**% Listagem comentada:**

```

%           Obs.: A dimensão de b e c pode ser um ou dois dependendo da quantidade de
%           regiões que têm soluções.

```

```

if multiponto=1

```

```

    % Esta condicional evita que sempre passe por este ponto, no levantamento da
    % curva Pdrop=f(M).

```

```

op031=menu('Escolha o tipo de mapeamento das regiões',
    '[Região inteira]',
    '[Subdividir a região em áreas unitárias]');

```

```

op03=op031 - 1

```

```

end

```

```

% Chama a subrotina reg01 para mapear as funções na
% região 01=> (0 < b < 1) e (0 < c < 1)

```

**reg01**

```

% Chama a subrotina reg01 para mapear as funções na
% região 02=> (0 < b < 1) e (0 < c < 10)

```

**reg02**

```

% Chama a subrotina reg03 afim de mapear a região 03
% onde (1 < b < Md) e ( c < 1)

```

**reg03**

```

% Chama a subrotina reg4 afim de mapear a região 04
% onde (1 < b < 10) e ( 1 < c < 10)

```

**reg04**

```

% Chama a subrotina abaixo para calcular a probabilidade de perda de pacotes de voz.

```

**probperd**

**A1.6****%Nome da subrotina: REG01****%Descrição:**

% Esta subrotina é chamada por VALIN e tem a função de mapear a região 01 que  
 % corresponde a encontrar os valores de  $f(b,c)$  e  $f_i(b,c)$ , onde o par ordenado  
 %  $(b,c)$  pertence ao seguinte intervalo:  $(0 < b < 1)$  e  $(0 < c < 1)$

**% Variáveis de entrada:**

São fornecidas pela subrotina DADOS1 a qual é chamada antes do mapeamento.

**% Variáveis de saída:**

%  $f$  :Matriz com todos os valores do mapeamento com dimensão (99 X 99).  
 %  $f_i$  :Matriz com todos os valores do mapeamento com dimensão (99 X 99).

**% Listagem comentada:**

```
clear b1
clear c1
contador=1;
% Chama-se a subrotina DADOS para carregar os dados
dados
região=1
% Constrói-se dois vetores b e c de tamanho 99 (cada) dentro da região
% pré-estabelecida
b=0.01:0.01:1;
c=0.01:0.01:1;
for i=1:1:99,
    for j=1:1:99,
        ce(i,j)=c(j);
        be(i,j)=b(i);
    end
end
% Mapeia-se as funções {f, fi} com os vetores acima encontrados.
u=(1 - PD).^be;
ud=(1 - p).^ce;
numr=p*ce.*u*(1 - g);
denr=numr + gf;
r=numr./denr;
f=h1*ce+h2*r - M;
fi=be.*ud*PD.*(1 - r)/sigd - Md;
clear be
clear ce
clear u
clear ud
clear numr
clear r
% Apesar de nesta região ter-se sempre área unitária toma-se o seguinte valor de min.
min=1;
% Chama-se a subrotina MAT para tentar encontrar os valores de B0 e C0
mat
```

**A1.7****% Nome da subrotina: DADOS****% Descrição:**

% Esta subrotina é chamada pelas subrotinas REG01, REG02, REG03 e REG04 e tem a %  
função de adequar os dados corretos para a opção escolhida dentre as várias oferecidas.

**% Variáveis de entrada:**

% As variáveis já estão disponíveis no programa a menos que o usuário deseje  
% modificar.

% t1 :Duração média de um surto de voz.  
% t2 :Duração média de um intervalo de silêncio.  
% t3 :Duração média de um minisurto.  
% t4 :Duração média de um minisilêncio.  
% Rc :Taxa do canal de comunicação em bits/s.  
% Rs :Taxa da fonte.  
% Dmax :Máximo atraso permitido para voz na transmissão.  
% T :Duração do "Frame".  
% M :Número de terminais de voz.

**% Variáveis de saída:**

% N :Número de pacotes por "Frame".  
% tau :Duração de um "slot".  
% g :Gama é a probabilidade que um surto de voz, com duração média t1 s  
% termine num "slot" de duração média "tau" s.  
% gf :Probabilidade que o surto termine no "frame" mais recente.  
% gm :Probabilidade que um minisurto termine num "slot" de tempo.  
% gmf :Probabilidade que um minisurto de voz termine no "frame" mais recente.  
% sig :Probabilidade de que num intervalo de silêncio, de duração média t2 s,  
% termine num "slot" de duração média tau s.  
% sigm :Probabilidade de haver uma transição do SIL para TALK.  
% alfa :Probabilidade de um minisurto ser o último no seu surto principal.  
% B :Capacidade da memória (pacotes)  
% h1 :Constante  
% h2 :Constante

**% Listagem comentada:**

```
%-----Características do terminal de Voz-----
t1=1.0;
t2=1.35;
t3=0.275;
t4=0.050;
Rc=720000;
Rs=32000;
T=0.016;
H=64;
Dmax=0.032;
%-----Características dos terminais de Dados -----
if especMd=1
    Md=Mpj;
else
    Md=M;
end
Rd=1200;
N=ceil(Rc * T/(Rs*T + H));
sigd=Rd/(Rs*N);
tau=T/N;
```

```

g=1 - exp(-tau/t1);
gf=1 - (1 - g).^N;
gm=1 - exp(-tau/t3);
gmf=1 - (1 - gm).^N;
sig=1 - exp(-tau/t2);
sigm=1 - exp(-tau/t4);
alfa=(t3 + t4)/t1;
B=ceil(Dmax/T);
D=floor(Dmax/tau);
h1=1 + g/sig;
h2=N + gf/sig;
Mdx=Md;
Mx=20;

```

### A1.8

**% Nome da subrotina: MAT**

**% Descrição:**

```

%      Esta subrotina é chamada por REG01.m, REG02, REG03 e REG04, encontra as matrizes
%      A11 (2xM) e A22 (2xN) cujas colunas são os pares ordenados (x,y) que tornam
%      f(x,y)=0 e fi(x,y)=0.

```

**% Variáveis de entrada:**

```

%      f(b,c) :Matriz de dimensão (bXc) com todos os valores mapeados da função.
%      fi(b,c) :Matriz de dimensão (bXc) com todos os valores mapeados da função.

```

**% Variáveis de saída:**

```

%      A11
%      A22

```

**% Listagem comentada:**

```

%      zx:      Identifica se uma das matrizes A11 ou A22 não é nula.
%      Cálculo o valor absoluto de F(b,c) e de Fi(b,c)
abf=abs(f);
abfi=abs(fi);
clear f
clear fi
% Deve-se estabelecer dois limiares afim de formar-se duas matrizes reduzidas de abf e abfi
% estes limiares diferem de acordo com o tipo de mapeamento; se o mapeamento for em área
% unitária o limiar será menor, caso contrário será maior.
if min=0
    limiar=0.1;
else
    limiar=0.8;
end
% As matrizes reduzidas A11 e A22 têm elementos cujos pares ordenados (b,c) são tais que
% tornam os valores absolutos das funções {f, fi} menores que o limiar previamente estabelecido.
% Encontra-se a matriz A11(2X?) com os pares ordenados (B, C) cujos valores posteriormente
% acertados, e substituídos em f(b,c), resultam em abf(b,c) < limiar.
[n1, m1]=find(abf < limiar);
clear A11
A11=[n1, m1]';
A12=A11'*A11;
dima=length(A12);
clear abf

```

```

clear A12
% Encontra a matriz A22(2X?) com os pares ordenados (B, C) cujos valores posteriormente
% acertados, e substituídos em fi(b,c) resultam em abfi(b,c) < limiar.
[n2, m2]=find(abfi < limiar);
clear A22;
A22=[n2, m2]';
A23=A22'* A22;
dimb=length(A23);
clear abfi
clear A23
% Se zx=0 -> ou as duas matrizes A11 e A22 não contêm elementos ou uma delas não contêm elementos.
zx=dima * dimb;
mat1

```

### A1.9

**% Nome da subrotina: MAT1**

**% Descrição:**

% Esta subrotina é chamada por MAT e tem a função de encontrar os vetores solução do  
 % sistema de equações  $f(b,c)=0$  e  $fi(b,c)=0$ .

**% Variáveis de entrada:**

% zx :Constante que verifica se a matriz A11 é vazia ou não.  
 % A11 :Matriz com os valores dos pares ordenados (b,c) que tornam  
 % simultaneamente  $fi(b,) < \text{limiar}$  e  $f(b,c) < \text{limiar}$ .

**% Variáveis de saída:**

% bx1=[b1 b2 b3 b4]:Vetor solução bx.  
 % cx1=[c1 c2 c3 c4]:Vetor solução cx.

**% Listagem comentada:**

```

% Os vetores bx1 e cx1 podem ter até quatro soluções.
if zx > 0
% Chama-se a subrotina COMP para comparar se existem colunas iguais entre A11 e A22
% Caso exista esta é a aproximação inicial de b e c.
    comp
    if contador=1
        comprimento=length(x10);
% Verifica se existem colunas em comum entre A11 e A22.
% Se comprimento > 0 tem-se elementos em comum entre A11 e A22.
        if comprimento=0
% Caso não encontre elementos comuns entre A11 e A22 os vetores b1 e c1 devem ser vazios
% afim de que não alterem os valores dos vetores bx1 e cx1.
            b1=[ ];
            c1=[ ];
            i1=[ ];
        elseif comprimento > 0
% Caso existam valores iniciais b0 e c0 deve-se recorrer ao método de Newton Raphson para encontrar
% o par ordenado (b,c) que é solução do sistema de equações.
% {raizdad1,raizdad2, raizdad3, raizdad4} são as quatro subrotinas que calculam as raízes para
% as diversas regiões.
            raizdad1
        end
    elseif contador=2
        comprimento=length(x20);

```



```

        if comprimento==0
            b2=[ ];
            c2=[ ];
            fi2=[ ];
        elseif comprimento > 0
            raizdad2
        end
    elseif contador=3
        comprimento=length(x30);
        if comprimento==0
            b3=[ ];
            c3=[ ];
            fi3=[ ];
        elseif comprimento > 0
            raizdad3
        end
    elseif contador=4
        comprimento=length(x40);
        if comprimento==0
            b4=[ ];
            c4=[ ];
            fi4=[ ];
        elseif comprimento > 0
            raizdad4
        end
    end
    dimx0=comprimento + 1;
    if dimx0 > 1
        clear bx1
        clear cx1
        bx1=[b1 b2 b3 b4];
        cx1=[c1 c2 c3 c4];
    end
end
end

```

#### A1.10

% Nome da subrotina: **COMP.m**

% **Descrição:**

% Esta subrotina é chamada por MAT1; ela compara duas matrizes de dimensões (2xM) e (2xN) e encontra as colunas iguais entre elas.

% **Variáveis de entrada:**

% Matriz A22

% Matriz A11

% **Variáveis de saída:**

% x0

% y0

% **Listagem comentada:**

% dima: é a dimensão da matriz A

% dimb: é a dimensão da matriz B

clear x0 % Limpa o valor anterior

clear y0 % Limpa o valor anterior

```

for k=1:1:dima
    for l=1:1:dimb;
%       X11 compara as duas primeiras linhas de A11 e A22
%       X22 compara as segundas linhas de A11 e A22
            x11(k,l)=abs( A11(1,k) - A22(1,l));
            x22(k,l)=abs( A11(2,k) - A22(2,l));
    end
end
% res é uma matriz quadrada cujos elementos nulos são aqueles onde a solução existe.
res=x11 + x22;
clear x11
clear x22
[z, w]=find(res=0);
% [z] é a posição na matriz A11 onde tem-se a solução aproximada do sistema de equações.
z;
if contador=1
    x0=A11(1,z)/100;
    y0=A11(2,z)/100;
    x10=x0;
elseif contador=2
    x0=A11(1,z)/100;
    y0=1 + A11(2,z)/10;
    x20=x0;
elseif contador=3
    x0=1 + A11(1,z)/10 ;
    y0=A11(2,z)/100;
    x30=x0;
elseif contador=4
    x0=1 + A11(1,z)/10;
    y0=1 + A11(2,z)/10;
    x40=x0;
end
end

```

### A1.11

**% Nome da subrotina: RAIZDAD1**

**% Descrição:**

% Esta subrotina é chamada por MAT1 e tem a finalidade de solicitar o cálculo de todas  
 % as raízes dos vetores - pontos iniciais x0 e y0 - e escolher aquelas que realmente  
 % são diferentes dentro de uma das quatro regiões em estudo.

**% Variáveis de entrada:**

% (X01, y01) :Par ordenado solução aproximada inicial para o cálculo do par  
 % ordenado (b1,c1) por um método adequado.

**% Variáveis de saída:**

% Par ordenado (b1, c1) que é a solução do sistema de equações  $f_1(b,c)=0$  e  $f_2(b,c)=0$ .

**% Listagem comentada:**

```

compx0=length(x0);
compy0=length(y0);
x01=x0;
y01=y0;
for var=1:1:compx0,
    b1=x01(var);

```

```

        c1=y01(var) ;
        nr1nr
%-----sol(.) -> Significa vetor solução da variável (.)
        solb1(var)=b1;
        solc1(var)=c1;
        solfi(var)=fi1;
        solf(var)=f1;
% Escolhe-se a(s) raiz(es) dentro da mesma região verificando se houve diferença significativa
% entre as componentes de cada vetor (solb) e (solc). Ver exemplo abaixo.
% Exemplo:
% Dado o vetor a=[.1; .4; .2; 1.2; 1.3; 1.5]
% Sendo delta=.5 por exemplo tem-se duas soluções possíveis:
% Uma entre .1 a .4 e outra que poderá ser 1.2, 1.3 ou 1.5.
% Fazendo este processo assegura-se que na ocorrência de duas raízes dentro da mesma região
% pode-se localizá-las sem problemas.
% -----Para encontrar b e c -----
        k=length(solb1) - 1;
        delta=.5;
        ib1=1;
        for x=1:1:k
                rse(x)=abs( solb1(x+1) - solb1(x));
                if rse(x) < delta
                        valorb1(ib1)=solb1(x);
                        valorc1(ib1)=solc1(x);
                elseif rse(x) > delta
                        valorb1(ib1+1)=solb1(x+1);
                        valorc1(ib1+1)=solc1(x+1);
                        ib1=ib1+ 1;
                end
        end
end
clear k
b1=valorb1;
c1=valorc1;
% Teste de verificação das raízes.
u=(1 - PD)^b1;
ud=(1 - p)^c1;
numr=c1*p*u*(1 - g);
denr=numr + gf;
r=numr/denr;
f1=h1*c1 + h2*r - M;
fi1=b1*ud*PD*(1 - r)/sigd - Md;

```

### A1.12

% **Nome da subrotina: NR1NR**

% **Descrição:**

% Esta subrotina é chamada por RAIZDAD1.m e tem a finalidade de encontrar a solução  
 % do sistema de equações  $f(b,c)=0$  e  $fi(b,c)=0$ , dentro da região  $c < 1$  e  $b < 1$ .

% **Variáveis de entrada:**

% Par ordenado (b0, c0).

% **Variáveis de saída:**

% Par ordenado (b1, c1) que é a solução do sistema de equações.

```

% Listagem comentada:
% ----- Entrada dos valores iniciais c0 e b0 já encontrados -----
clear b
clear c
b=b1;
c=c1;
h1=1 + g/sig;
h2=N + gf/sig;
kd=PD/sigd;
u=(1 - PD)^b ;
ud=((1 - p)^c);
numr=c*p*u*(1 - g);
denr=numr + gf;
r=numr/denr;
% Como o algoritmo pode não convergir o aplicativo ficaria num "loop" infinito. Para
% evitar este problema cria-se um contador de iterações e caso não haja convergência antes de
% atingir este valor o contador pára os cálculos e o aplicativo admite os valores iniciais
% de entrada como solução do problema.
contalgo=0;
% Contalgo - Contador do algoritmo.
% Adota-se um erro inicial.
erro=10;
while erro > 0.0001
    deri1
    u=(1 - PD)^b ;
    ud=((1 - p)^c);
    numr=c*p*u*(1 - g);
    denr=numr + gf;
    r=numr/denr;
    f=h1*c + h2*r - M;
    fi=b*ud*PD*(1 - r)/sigd - Md;
    b01=b + (fi*DfDc - f *DfiDc)/Jacob;
    c01=c + (f*DfiDb - fi * DfDb)/Jacob;
    erro=abs(f) + abs(fi) ;
    c=c01;
    b=b01;
% Contador de iterações do algoritmo
    contalgo=contalgo + 1;
    if contalgo=300
        erro=0.0001;
        b=[ ];
        c=[ ];
    end
end
b1=b;
c1=c;
clear b
clear c
u=((1 - PD)^b1);
ud=((1 - p)^c1);
numr=c1*p*u*(1 - g);
denr=numr + gf;
r=numr/denr;
f1=h1*c1 + h2 *r - M;
fi1=b1*ud*PD*(1 - r)/sigd - Md;

```

**A1.13****% Nome da subrotina: DERI1****% Descrição:**

% Esta subrotina é chamada por NR1NR.m e encontra os valores das derivadas parciais  
% e o Jacobiano necessários para o cálculo pelo método de Newton Raphson.

**% Variáveis de entrada:**

% c :Número de terminais de voz em disputa no equilíbrio.  
% b :Número de terminais de dados em disputa no equilíbrio.  
% Md :Número de terminais de dados.  
% g :Gama é a probabilidade de que um surto de voz, com duração média t1 s  
% termine num "slot" de duração média "tau" s.  
% gf :Probabilidade que o surto termine no "frame" mais recente.  
% sigd :Probabilidade de que um pacote seja gerado num terminal em algum "slot"  
% h1 :Constante  
% h2 :Constante  
% PD :Probabilidade de acesso de dados.  
% p :Probabilidade de acesso de voz.

**% Variáveis de saída:**

% DfDb :derivada de f(b,c) em relação a **b**.  
% DfDc :derivada de f(b,c) em relação a **c**.  
% DfiDb :derivada de fi(b,c) em relação a **b**.  
% DfiDc :derivada de fi(b,c) em relação a **c**.  
% Jocaob :Jacobiano.

**% Listagem comentada:**

% Obs. Dx Dy -> Derivada de x em relação a y.  
% Cálculo da derivada f'(b,c) em relação a **b**  
DF1=c\*(1 - g)\*p\*((1 - PD).^b)\*log(1 - PD);  
DF2=gf + c\*(1 - g)\*p\*((1 - PD).^b) ;  
DF3=((c\*(1 - g)\*p\*((1 - PD).^b)).^2)\*log(1 - PD);  
DF4=DF2.^2;  
DrDb=( DF1/DF2 - DF3/(DF4));  
DfDb=h2 \* DrDb;  
% Cálculo da derivada f'(b,c) em relação a **c**  
DF1=(1 - g)\*p\*((1 - PD).^b);  
DF2=gf + c\*DF1;  
DF3=c\*DF1;  
DrDc=(DF1/DF2)\*(1 - DF3/DF2);  
DfDc=h1 + h2\*DrDc;  
% Cálculo da derivada fi'(b,c) em relação a **b**  
DudDb=0;  
DF1=c\*(1 - g)\*p\*((1 - PD).^b)\*log(1 - PD);  
DF2=gf + c\*(1 - g)\*p\*((1 - PD).^b) ;  
DF3=c\*(1 - g)\*p\*((1 - PD).^b);  
DrDb=( DF1/DF2)\*(1 - DF3/DF2);  
DfiDb=b\*kd\*DudDb - kd\*b\*(DudDb\*r + ud\*DrDb)+kd\*ud - kd\*ud\*r;  
% Cálculo da derivada fi'(b,c) em relação a **c**  
DudDc=((1 - p)^c)\*log(1 - p);  
DF1=(1 - g)\*p\*((1 - PD).^b);  
DF2=gf + c\*DF1;  
DF3=c\*DF1;  
DrDc=(DF1/DF2)\*(1 - DF3/DF2);  
DfiDc=b\*kd\*DudDc - kd\*b\*(DudDc\*r + ud\*DrDc);  
Jacob=det([DfDb DfDc; DfiDb DfiDc]);

**A1.14****% Nome da subrotina: RAIZDAD2****% Descrição:**

% Esta subrotina é chamada por MAT1 e tem a finalidade de solicitar o cálculo de todas  
 % as raízes dos vetores - pontos iniciais x0 e y0 - e escolher aquelas que realmente  
 % são diferentes dentro de uma das quatro regiões em estudo.

**% Variáveis de entrada:**

% (X02, y02) :Par ordenado solução aproximada inicial para o cálculo do par  
 % ordenado (b2,c2) por um método adequado.

**% Variáveis de saída:**

% Par ordenado (b2, c2) que é a solução do sistema de equações  $f_1(b,c)=0$  e  $f_2(b,c)=0$ .

**% Listagem comentada:**

```
compx0=length(x0);
compy0=length(y0);
x0t=x0;
y0t=y0;
x02=x0;
y02=y0;
for var=1:1:compx0,
    b2=x0(var);
    c2=y0t(var) ;
    nr2nr
%-----sol(.) -> Significa vetor solução da variável (.)
    solb(var)=b2;
    solc(var)=c2;
    solfi(var)=fi2;
    solf(var)=f2;
%-----COMENTÁRIO-----
%---Escolhe-se a(s) raiz(es) dentro da mesma região verificando se houve diferença significativa
%---entre as componentes de cada vetor (solb) e (solc). Ver exemplo abaixo.
% Exemplo:
% Dado o vetor a=[.1; .4; .2; 1.2; 1.3; 1.5]
% Sendo delta=.5 por exemplo tem-se duas soluções possíveis:
% Uma entre .1 a .4 e outra que poderá ser 1.2, 1.3 ou 1.5.
% Fazendo este processo assegura-se que na ocorrência de duas raízes dentro da mesma região
% pode-se localizá-las sem problemas.
% -----Para encontrar-se b e c -----
% Necessita-se colocar esta condicional ( if length(solb) > 1 ) porque se
% o comprimento de solb for igual a um haverá problemas com o "k".
    if length(solb) > 1
        k=length(solb) - 1;
        delta=.5;
        ib=1;
        for x=1:1:k
            rse(x)=abs( solb(x+1) - solb(x));
            if rse(x) < delta
                valorb(ib)=solb(x);
                valorc(ib)=solc(x);
            elseif rse(x) > delta
                valorb(ib+1)=solb(x+1);
                valorc(ib+1)=solc(x+1);
                ib=ib + 1;
            end
        end
    end
```

```

end
end
end
if length(solb) > 1
    b2=valorb;
    c2=valorc;
end
clear solb
clear solc
clear solfi
clear solf
clear valorb
clear valorc
% Teste de verificação das raízes.
u=((1 - PD)^b2)*((1 - p)^(c2 - 1)) ;
ud=((1 - p)^c2)*((1 - PD)^(b2 - 1));
denr=numr + gf;
r=numr/denr;
f2=h1*c2 + h2 *r - M
fi2=b2*ud*PD*(1 - r)/sigd - Md

```

### A1.15

**% Nome da subrotina: NR2NR**

**% Descrição:**

% Esta subrotina é chamada por RAIZDAD2.m e tem a finalidade de encontrar a solução  
 % do sistema de equações  $f(b,c)=0$  e  $fi(b,c)=0$ , dentro da região  $c > 1$  e  $b < 1$ .

**% Variáveis de entrada:**

% Par ordenado (b0, c0).

**% Variáveis de saída:**

% Par ordenado (b2, c2) que é a solução do sistema de equações.

**% Listagem comentada:**

% ----- Entrada dos valores iniciais c0 e b0 já encontrados -----

```

clear b
clear c
b=b2;
c=c4;
h1=1 + g/sig;
h2=N + gf/sig;
kd=PD/sigd;
u=((1 - p)^(c - 1))*(1 - PD)^b ;
ud=((1 - p)^c);
numr=c*p*u*(1 - g);
denr=numr + gf;
r=numr/denr;
% Como o algoritmo pode não convergir o aplicativo ficaria num "loop" infinito. Para
% evitar este problema cria-se um contador de iterações e, caso não haja convergência antes de
% atingir este valor o contador pára os cálculos e o aplicativo admite os valores iniciais
% de entrada como solução do problema.
contalgo=0;
% Contalgo - Contador do algoritmo.
% Adota-se um erro inicial.

```

```

erro=10;
while erro > 0.0001
    deri2
    u=((1 - p)^(c - 1))*(1 - PD)^b ;
    ud=((1 - p)^c);
    numr=c*p*u*(1 - g);
    denr=numr + gf;
    r=numr/denr;
    f=h1*c + h2*r - M;
    fi=b*ud*PD*(1 - r)/sigd - Md;
    b02=b + (fi*DfDc - f * DfiDc)/Jacob;
    c02=c + (f*DfiDb - fi * DfDb)/Jacob;
    erro=abs(f) + abs(fi) ;
    c=c02;
    b=b02;
% Contador de iterações do algoritmo
    contalgo=contalgo + 1;
    if contalgo=1500;
        erro=0.0001;
        b=[ ];
        c=[ ];
    end
end
b2=b
c2=c
clear b
clear c
u=((1 - PD)^b2)*((1 - p)^(c2 - 1)) ;
ud=((1 - p)^c2)*((1 - PD)^(b2 - 1));
numr=c2*p*u*(1 - g);
denr=numr + gf;
r=numr/denr;
f2=h1*c2 + h2 *r - M
fi2=b2*ud*PD*(1 - r)/sigd - Md

```

### A1.16

% Nome da subrotina: **DERI2**

% **Descrição:**

% Esta subrotina é chamada por NR2NR.m e encontra os valores das derivadas parciais  
 % e o Jacobiano necessários para o cálculo pelo método de Newton Raphson.

% **Variáveis de entrada:**

%	c	:Número de terminais de voz em disputa no equilíbrio.
%	b	:Número de terminais de dados em disputa no equilíbrio.
%	Md	:Número de terminais de dados.
%	g	:Gama é a probabilidade que um surto de voz, com duração média t1 s
%		termine num "slot" de duração média "tau" s.
%	gf	:Probabilidade que o surto termine no "frame" mais recente.
%	sigd	:Probabilidade de que um pacote seja gerado num terminal em algum "slot"
%	h1	:Constante
%	h2	:Constante
%	PD	:Probabilidade de acesso de dados
%	p	:Probabilidade de acesso de voz.



**% Variáveis de saída:**

```
%
      DfDb   :derivada de f(b,c) em relação a b.
%
      DfDc   :derivada de f(b,c) em relação a c.
%
      DfiDb  :derivada de fi(b,c) em relação a b.
%
      DfiDc  :derivada de fi(b,c) em relação a c.
%
      Jocaob :Jacobiano.
```

**% Listagem comentada:**

```
% Obs. Dx Dy -> Derivada de x em relação a y.
% Cálculo da derivada f(b,c) em relação a b
DF1=c*(1 - g)*((1 - p).^(c - 1))*p*((1 - PD).^b)*log(1 - PD);
DF2=gf + c*(1 - g)*((1 - p).^(c - 1))*p*((1 - PD).^b) ;
DF3=((c*(1 - g)*((1 - p).^(c - 1))*p*((1 - PD).^b)).^2)*log(1 - PD);
DF4=DF2.^2;
DrDb=( DF1/DF2 - DF3/(DF4));
DfDb=h2 * DrDb;
% Cálculo da derivada f(b,c) em relação a c
DF1=(1 - g)*((1 - p).^(c - 1))*p*((1 - PD).^b);
DF2=gf + c*DF1;
DF3=c*DF1*log(1 - p);
DF4=c*DF1;
DF5=DF2.^2;
DrDc=((DF1 + DF3)/DF2)*(1 - DF4/DF2);
DfDc=h1 + h2*DrDc;
% Cálculo da derivada fi'(b,c) em relação a b
DudDb=0 ;
DF1=c*(1 - g)*((1 - p).^(c - 1))*p*((1 - PD).^b) ;
DF2=gf + DF1;
DF3=c*(1 - g)*((1 - p).^(c - 1))*p*((1 - PD).^b)*log(1 - PD);
DrDb=(DF3/DF2)*(1 - DF1/DF2);
DfiDb=b*kd*DudDb - kd*b*(DudDb*r + ud*DrDb)+kd*ud - kd*ud*r;
% Cálculo da derivada fi'(b,c) em relação a c
DudDc=((1 - p)^c)*log(1 - p);
DF1=(1 - g)*((1 - p).^(c - 1))*p*((1 - PD).^b);
DF2=gf + c*DF1;
DF3=c*DF1*log(1 - p);
DF4=c*DF1;
DrDc=((DF1 + DF3) /DF2)*(1 - DF4/DF2);
DfiDc=b*kd*DudDc - kd*b*(DudDc*r + ud*DrDc);
Jacob=det([DfDb DfDc; DfiDb DfiDc]);
```

**A1.17****% Nome da subrotina: RAIZDAD3****% Descrição:**

```
%
      Esta subrotina é chamada por MAT1 e tem a finalidade de solicitar o cálculo de todas
%
      as raízes dos vetores - pontos iniciais x0 e y0 - e escolher aquelas que realmente
%
      são diferentes dentro de uma das quatro regiões em estudo.
```

**% Variáveis de entrada:**

```
%
      (X03, y03)   :Par ordenado solução aproximada inicial para o cálculo do par
%
                   ordenado (b3,c3) por um método adequado.
```

**% Variáveis de saída:**

```

%                               Par ordenado (b3, c3) que é a solução do sistema de equações  $f_i(b,c)=0$  e  $f(b,c)=0$ .

% Listagem comentada:
clear compx0
clear compy0
compx0=length(x0);
compy0=length(y0);
x03=x0;
y03=y0;
for var=1:1:compx0,
    b3=x03(var);
    c3=y03(var) ;
% Afim de diminuir o tempo de cálculo chama-se a subrotina
% nr3 que calcula pelo método iterativo.
    nr3
%-----sol(.) -> Significa vetor solução da variável (.)
% Afim de evitar soluções negativas faz-se:
    if b3 < 0
        solb3(var)=[ ];
        solc3(var)=[ ];
    elseif c3 < 0
        solb3(var)=[ ];
        solc3(var)=[ ];
    else
        solb3(var)=b3;
        solc3(var)=c3;
        solfi(var)=fi3;
        solf(var)=f3;
    end
%-----COMENTÁRIO-----
%---Escolhe-se a(s) raiz(es) dentro da mesma região verificando se houve diferença significativa
%---entre as componentes de cada vetor (solb) e (solc). Ver exemplo abaixo.
% Exemplo:
% Dado o vetor a=[.1; .4; .2; 1.2; 1.3; 1.5]
% Sendo delta=.5 por exemplo tem-se duas soluções possíveis:
% Uma entre .1 a .4 e outra que poderá ser 1.2, 1.3 ou 1.5.
% Fazendo este processo assegura-se que na ocorrência de duas raízes dentro da mesma região
% pode-se localizá-las sem problemas.
end
clear solb
clear solc
clear solfi
clear solf
clear valorb
clear valorc
% Teste de verificação das raízes.
h1=1 + g/sig;
h2=N + gf/sig;
kd=PD/sigd;
u=(1 - PD)^b3 ;
ud=((1 - p)^c3)*((1 - PD)^(b3 - 1));
numr=c3*p*u*(1 - g);
denr=numr + gf;
r=numr/denr;
f3=h1*c3 + h2 *r - M;
fi3=b3*ud*PD*(1 - r)/sigd - Md;

```

**A1.18****% Nome da subrotina: NR3****% Descrição:**

% Esta subrotina é chamada por RAIZDAD2.m e tem a finalidade de encontrar a solução  
 % do sistema de equações  $f(b,c)=0$  e  $f_i(b,c)=0$ , dentro da região  $1 < c$  e  $b > 1$ .

**% Variáveis de entrada:**

% Par ordenado (b0, c0).

**% Variáveis de saída:**

% Par ordenado (b3, c3) que é a solução do sistema de equações.

**% Listagem comentada:**

% ----- Entrada dos valores iniciais c0 e b0 já encontrados -----

% Como x0 e y0 são dois vetores, substitui-se o valor médio de suas componentes  
 % como pontos iniciais no cálculo das raízes.

clear c

clear b

b=b3;

c=c3;

% Cálculo da derivada de  $f(b,c)$  em relação a **b**

DF1=c\*(1 - g)\*h2\*p\*((1 - PD)^b)\*log10(1 - PD);

DF2=c\*(1 - g)\*p\*((1 - PD).^b) + gf;

DF3=DF1\*c\*(1 - g)\*p\*(1 - PD);

Dfbcb=DF1/DF2 - DF3/(DF2^2);

% Cálculo da derivada  $f(b,c)$  em relação a **c**

DF1=(1 - g)\*h2\*p\*((1 - PD)^b);

DF2=gf + c\*(1 - g)\*p\*((1 - PD)^b);

DF3=c\*(1 - g)\*p\*((1 - PD)^b);

DF4=DF3\*DF1/(DF2^2);

Dfbcc=h1 + DF1/DF2 - DF4;

% Cálculo da derivada  $f_i(b,c)$  em relação a **b**

DF1=c\*(1 - g)\*p\*((1 - PD)^b);

DF2=gf + c\*(1 - g)\*p\*((1 - PD)^b);

DF3=((1 - p)^c)\*((1 - PD)^(b - 1))\*PD;

DF4=(1 - DF1/DF2);

DF5=DF3\*DF4/sigd;

DF6=DF5\*(1 + b\*log10(1 - PD));

DF7=DF1\*log10(1 - PD)/DF2;

DF8=(b \* DF3 \* (1 - DF1/DF2) \* log10(1 - PD)) / sigd;

DF9=(b \* DF3 \* DF7 \* (-1 + DF1/DF2) ) / sigd;

Dfibcb=DF5 + DF8 + DF9;

% Cálculo da derivada  $f_i(b,c)$  em relação a **c**

DF1=c\*(1 - g)\*p\*((1 - PD)^b);

DF2=gf + c\*(1 - g)\*p\*((1 - PD)^b);

DF3=b\*((1 - p)^c);

DF4=(1 - DF1/DF2);

DF5=(1 - PD)^(b - 1);

DF6=(1 - g)\*p\*((1 - PD)^b);

DF7=(-DF6/DF2 + DF1 \* DF6\*DF5\*PD/DF2^2);

DF8=DF3 \* (1 - DF1/DF2) \* DF5 \* PD \* log10(1 - p) / sigd;

Dfibcc=(DF3 \* (- DF6/DF2 + c\*((DF6/DF2)^2) ) \* DF5 \* PD ) / sigd + DF8;

% ---Cálculo da Raiz -----

AM=[Dfbcb Dfibcb; Dfbcc Dfibcc];

BM=[ -1 Dfibcb; 0 Dfibcc];

```

CM=[Dfbc b -1 ; Dfbcc 0 ];
DM=[ 0 Dfibcb; -1 Dfibcc];
EM=[Dfbc b 0 ; Dfbcc -1 ];
alfa=det(BM)/det(AM);
beta=det(CM)/det(AM);
gama=det(DM)/det(AM);
delta=det(EM)/det(AM);
b0=b;
c0=c;
% Afim de parar o processo caso o algoritmo não convirja lança-se mão de um novo contador.
contalgo=0;
% Contalgo - Contador do algoritmo
% Adota-se um erro inicial
erro=10;
while erro > 0.0001
    u=(1 - PD).^b0 ;
    ud=((1 - p)^c0)*((1 - PD)^(b0 - 1)) ;
    numr=c0*p*u*(1 - g);
    denr=numr + gf;
    r=numr/denr;
    f0=h1*c0 + h2*r - M;
    fi0=b0*ud*PD*(1 - r)/sigd - Md;
    b=b0 + alfa*f0 + beta*fi0;
    c=c0 + gama*f0 + delta*fi0;
    erro=abs(f0) + abs(fi0);
    c0=c;
    b0=b;
% Contador de iterações do algoritmo
    contalgo=contalgo + 1;
    if contalgo=300
        erro=0.0001;
        b0=[ ];
        c0=[ ];
    end
end
c3=c;
b3=b;
u=(1 - PD)^b3 ;
ud=((1 - p)^c3)*((1 - PD)^(b3 - 1)) ;
numr=c3*p*u*(1 - g);
denr=numr + gf;
r=numr/denr;
f3=h1*c3 + h2*r - M;
fi3=b3*ud*PD*(1 - r)/sigd - Md;

```

#### A1.19

**% Nome da subrotina: RAIZDAD4**

**% Descrição:**

% Esta subrotina é chamada por MAT1 e tem a finalidade de solicitar o cálculo de todas  
 % as raízes dos vetores - pontos iniciais x0 e y0 - e escolher aquelas que realmente  
 % são diferentes dentro de uma das quatro regiões em estudo.

**% Variáveis de entrada:**

% (X04, y04) :Par ordenado solução aproximada inicial para o cálculo do par

```

%                               ordenado (b3,c3) por um método adequado.

% Variáveis de saída:
%                               Par ordenado (b4, c4) que é a solução do sistema de equações  $f_i(b,c)=0$  e  $f(b,c)=0$ .

% Listagem comentada:
compX0=length(x0);
compY0=length(y0);
x0t=x0;
y0t=y0;
x04=x0;
y04=y0;
for var=1:1:compX0,
    b4=x0(var);
    c4=y0t(var) ;
    nr4nr
    f4teste=f4;
    fi4teste=fi4;
%-----sol(.) -> Significa vetor solução da variável (.)
    solb(var)=b4;
    solc(var)=c4;
    solfi(var)=fi4;
    solf(var)=f4;
%---Escolhe-se a(s) raiz(es) dentro da mesma região verificando se houve diferença significativa
%---entre as componentes de cada vetor (solb) e (solc). Ver exemplo abaixo.
% Exemplo:
% Dado o vetor a=[.1; .4; .2; 1.2; 1.3; 1.5]
% Sendo delta=.5 por exemplo tem-se duas soluções possíveis:
% Uma entre .1 a .4 e outra que poderá ser 1.2, 1.3 ou 1.5.
% Fazendo este processo assegura-se que na ocorrência de duas raízes dentro da mesma região
% pode-se localizá-las sem problemas.
% -----Para encontrar-se b e c -----
% Necessita-se colocar esta condicional ( if length(solb) > 1 ) porque se
% o comprimento de solb for igual a um haverá problemas com o "k".
    if length(solb) > 1
        k=length(solb) - 1;
        delta=.5;
        ib=1;
        for x=1:1:k
            rse(x)=abs( solb(x+1) - solb(x));
            e(x) < delta
            valorb(ib)=solb(x);
            valorc(ib)=solc(x);
        elseif rse(x) > delta
            valorb(ib+1)=solb(x+1);
            valorc(ib+1)=solc(x+1);
            ib=ib + 1;
        end
    end
end
end
if length(solb) > 1
    b4=valorb;
    c4=valorc;
end
clear solb
clear solc

```

```

clear solfi
clear solf
clear valorb
clear valorc
% Teste de verificação das raízes.
u=((1 - PD)^b4)*((1 - p)^(c4 - 1)) ;
ud=((1 - p)^c4)*((1 - PD)^(b4 - 1));
denr=numr + gf;
r=numr/denr;
f4=h1*c4 + h2 *r - M;
fi4=b4*ud*PD*(1 - r)/sigd - Md;

```

## A1.20

**% Nome da subrotina: NR4NR**

**% Descrição:**

% Esta subrotina é chamada por RAIZDAD4.m e tem a finalidade de encontrar a solução  
 % do sistema de equações  $f(b,c)=0$  e  $fi(b,c)=0$ , dentro da região  $c > 1$  e  $b > 1$ .

**% Variáveis de entrada:**

% Par ordenado (b0, c0).

**% Variáveis de saída:**

% Par ordenado (b4, c4) que é a solução do sistema de equações.

**% Listagem comentada:**

% ----- Entrada dos valores iniciais c0 e b0 já encontrados -----

```

clear b
clear c
b=b4;
c=c4;
h1=1 + g/sig;
h2=N + gf/sig;
kd=PD/sigd;
u=((1 - p)^(c - 1))*(1 - PD)^b ;
ud=((1 - p)^c)*((1 - PD)^(b - 1));
numr=c*p*u*(1 - g);
denr=numr + gf;
r=numr/denr;

```

% Como o algoritmo pode não convergir então o aplicativo ficaria num "loop" infinito. Para  
 % evitar este problema cria-se um contador de iterações e, caso não haja convergência antes de  
 % atingir este valor o contador pára os cálculos e o aplicativo admite os valores iniciais  
 % de entrada como solução do problema.

```
contalgo=0;
```

% Contalgo - Contador do algoritmo.

% Adota-se um erro inicial.

```
erro=10;
```

```
while erro > 0.0001
```

**deri4**

```

u=((1 - p)^(c - 1))*(1 - PD)^b ;
ud=((1 - p)^c)*((1 - PD)^(b - 1));
numr=c*p*u*(1 - g);
denr=numr + gf;
r=numr/denr;
f=h1*c + h2*r - M;

```

```

    fi=b*ud*PD*(1 - r)/sigd - Md;
    b04=b + (fi*DfDc - f *DfiDc)/Jacob;
    c04=c + (f*DfiDb - fi * DfDb)/Jacob;
    erro=abs(f) + abs(fi) ;
    c=c04;
    b=b04;
% Contador de iterações do algoritmo
    contalgo=contalgo + 1;
    if contalgo=1500;
        erro=0.0001;
        b=[ ];
        c=[ ];
    end
end
b4=b
c4=c
clear b
clear c
u=((1 - PD)^b4)*((1 - p)^(c4 - 1)) ;
ud=((1 - p)^c4)*((1 - PD)^(b4 - 1));
numr=c4*p*u*(1 - g);
denr=numr + gf;
r=numr/denr;
f4=h1*c4 + h2 *r - M;
fi4=b4*ud*PD*(1 - r)/sigd - Md;

```

### A1.21

#### % Nome da subrotina: DERI4

#### % Descrição:

% Esta subrotina é chamada por NR4NR.m e encontra os valores das derivadas parciais  
 % e o Jacobiano necessários para o cálculo pelo método de Newton Raphson.

#### % Variáveis de entrada:

%	c	:Número de terminais de voz em disputa no equilíbrio.
%	b	:Número de terminais de dados em disputa no equilíbrio.
%	Md	:Número de terminais de dados.
%	g	:Gama é a probabilidade que um surto de voz, com duração média t1 s
%		termine num "slot" de duração média "tau" s.
%	gf	:Probabilidade que o surto termine no "frame" mais recente.
%	sigd	:Probabilidade de que um pacote seja gerado num terminal em algum "slot"
%	h1	:Constante
%	h2	:Constante
%	PD	:Probabilidade de acesso de dados.
%	p	:Probabilidade de acesso de voz.

#### % Variáveis de saída:

%	DfDb	:derivada de f(b,c) em relação a <b>b</b> .
%	DfDc	:derivada de f(b,c) em relação a <b>c</b> .
%	DfiDb	:derivada de fi(b,c) em relação a <b>b</b> .
%	DfiDc	:derivada de fi(b,c) em relação a <b>c</b> .
%	Jocaob	:Jacobiano.

#### % Listagem comentada:

```

% Obs. Dx Dy -> Derivada de x em relação a y.
% Cálculo da derivada f'(b,c) em relação a b
DF1=c*(1 - g)*((1 - p).^(c - 1))*p*((1 - PD).^b)*log(1 - PD);
DF2=gf + c*(1 - g)*((1 - p).^(c - 1))*p*((1 - PD).^b);
DF3=((c*(1 - g)*((1 - p).^(c - 1))*p*((1 - PD).^b)).^2)*log(1 - PD);
DF4=DF2.^2;
DrDb=( DF1/DF2 - DF3/(DF4));
DfDb=h2 * DrDb;
% Cálculo da derivada f'(b,c) em relação a c
DF1=(1 - g)*((1 - p).^(c - 1))*p*((1 - PD).^b);
DF2=gf + c*DF1;
DF3=c*DF1*log(1 - p);
DF4=c*DF1;
DF5=DF2.^2;
DrDc=((DF1 + DF3)/DF2)*(1 - DF4/DF2);
DfDc=h1 + h2*DrDc;
% Cálculo da derivada fi'(b,c) em relação a b
DudDb=((1 - p)^c)*((1 - PD)^(b - 1))*log(1 - PD);
DF1=c*(1 - g)*((1 - p)^(c - 1))*p*((1 - PD)^b);
DF2=gf + DF1;
DF3=c*(1 - g)*((1 - p).^(c - 1))*p*((1 - PD).^b)*log(1 - PD);
DrDb=(DF3/DF2)*(1 - DF1/DF2);
DfiDb=b*kd*DudDb - kd*b*(DudDb*r + ud*DrDb)+kd*ud - kd*ud*r;
% Cálculo da derivada fi'(b,c) em relação a c
DudDc=((1 - p)^c)*((1 - PD)^(b - 1))*log(1 - p);
DF1=(1 - g)*((1 - p).^(c - 1))*p*((1 - PD).^b);
DF2=gf + c*DF1;
DF3=c*DF1*log(1 - p);
DF4=c*DF1;
DrDc=(DF1/DF2) + (DF3 /DF2) - (DF4*(DF1 + DF3))/(DF2^2);
DfiDc=b*kd*DudDc - kd*b*(DudDc*r + ud*DrDc);
Jacob=det([DfDb DfDc; DfiDb DfiDc]);

```

## A1.22

%Nome da subrotina: REG02

### % Descrição:

```

%      Esta subrotina é chamada por VALIN e tem a função de mapear a região 02 que
%      corresponde a encontrar os valores de fi(b,c) e f(b,c), onde o par ordenado
%      (b,c) pertence ao seguinte intervalo: (0 < b < 1) e (1 < c < M)

```

### % Variáveis de entrada:

```

%      São fornecidas pela subrotina DADOS01 que é chamada antes do mapeamento.

```

### % Variáveis de saída:

```

%      f      :Matriz com todos os valores do mapeamento com dimensão (99 X 99).
%      fi     :Matriz com todos os valores do mapeamento com dimensão (99 X 99).

```

### % Listagem comentada:

```

% Limpa algumas variáveis afim de evitar conflito de dimensões de matrizes

```

```

    clear b1
    clear c1
    contador=2;

```

```

% Chama a subrotina DADOS01 para carregar os dados

```

```

    dados

```



```

        regioao=2
% Constrói-se dois vetores b e c dentro da região pré-estabelecida
        b=0.01:0.01:1;
        c=1.1:1:M;
        maxc=10*(Mx - 1.1);
% maxc - valor máximo de c.
        for i=1:1:99,
                for j=1:1:maxc,
                        ce(i,j)=c(j);
                        be(i,j)=b(i);
                end
        end
% Mapeia-se as funções {f, fi} com os vetores acima encontrados.
u=((1 - p).^(ce - 1)).*((1 - PD).^be);
ud=(1 - p).^ce;
numr=p*ce.*u*(1 - g);
denr=numr + gf;
r=numr./denr;
f=h1*ce+h2*r - M;
fi=be.*ud*PD.*(1 - r)/sigd - Md;
clear be
clear ce
clear u
clear ud
clear numr
clear r
% Apesar de sempre nesta região ter-se área unitária toma-se o seguinte valor de min.
min=1;
% Chama-se a subrotina MAT para tentar encontrar os valores de B0 e C0
mat

```

### A1.23

**% Nome da subrotina: REG03**

**% Descrição:**

% Esta subrotina é chamada por VALIN e tem a função de mapear a região 03 que  
 % corresponde a encontrar os valores de  $f_i(b,c)$  e  $f(b,c)$ , onde o par ordenado  
 %  $(b,c)$  pertence ao seguinte intervalo:  $(0 < b < Md)$  e  $(c < 1)$

**% Variáveis de entrada:**

% São fornecidas pela subrotina DADOS01 que é chamada antes do mapeamento.

**% Variáveis de saída:**

% f :Matriz com todos os valores do mapeamento com dimensão (99 X 99).

**% Listagem comentada:**

% Cálculo dos valores iniciais B0 e C0  
 % Limpa algumas variáveis afim de evitar conflito de dimensões de matrizes

**limpa**

contador=3;  
 % Chama a subrotina dados afim de carregá-los

**dados**

regiao=3  
 % Constrói-se dois vetores b e c dentro da região pré-estabelecida  
 b=1.1:1:Md;

```

maxb=10*(Mdx - 1.1);
% maxb -> valor máximo de b
% Md=1 pois começa em um
c=0.01:0.01:1;
for i=1:1:maxb,
    for j=1:1:99,
        ce(i,j)=c(j);
        be(i,j)=b(i);
    end
end
% Mapeia-se as funções {f, fi} com os vetores acima encontrados.
u=((1 - PD).^be);
ud=((1 - p).^ce).*((1 - PD).^(be - 1));
numr=p*ce.*u*(1 - g);
denr=numr + gf;
r=numr./denr;
f=h1*ce+h2*r - M;
fi=be.*ud*PD.*(1 - r)/sigd - Md;
clear be
clear ce
clear u
clear ud
clear numr
clear r
% Estabelece-se o valor de min=1 pois o mapeamento não é em área unitária
min=1;
% Chama a subrotina mat para calcular os valores de b0 e c0
mat
% O mapeamento em área unitária é feito subdividindo-se uma área qualquer
% num número inteiro de áreas unitárias e faz-se o mapeamento em cada
% sub-área separadamente.
% Faz-se o mapeamento em área unitária se:
% a - Foi escolhida a priori esta opção -> op03=1
% b - Durante o cálculo nesta área não unitária não foi encontrada raiz.
if op03=1
    detnull=length(fi3);
    if detnull=0
        % Faz-se o mapeamento por regiões menores com área unitária
        % Limpa algumas variáveis afim de evitar conflito de dimensões de matrizes
limpa
        contador=3;
        % Constrói-se dois vetores b e c dentro da região pré-estabelecida
        kvalini=1;
        clear ce
        clear be
        clear xvalini
        while kvalini < Mx;
            xvalini=kvalini + 1;
            ik=kvalini + .01;
            b=ik:.01:xvalini;
            c=0.01:.01:1;
            for i=1:1:99,
                for j=1:1:99,
                    ce(i,j)=c(j);
                    be(i,j)=b(i);
                end
            end
        end
    end
end

```

```

kvalini=kvalini + 1;
% Mapeia-se as funções {f, fi} com os vetores acima encontrados.
u=((1 - PD).^be);
ud=((1 - p).^ce).*((1 - PD).^(be - 1));
numr=p*ce.*u*(1 - g);
denr=numr + gf;
r=numr./denr;
f=h1*ce+h2*r - M;
fi=be.*ud*PD.*(1 - r)/sigd - Md;
clear u
clear ud
clear numr
clear r
mim=0;
% Chama a subrotina mat para calcular os valores de b0 e c0
mat
end
end
end

```

#### A1.24

**% Nome da subrotina: REG04**

**% Descrição:**

```

%      Esta subrotina é chamada por VALIN e tem a função de mapear a região 04 que
%      corresponde a encontrar os valores de fi(b,c) e f(b,c), onde o par ordenado
%      (b,c) pertence ao seguinte intervalo: (1 < b < Md) e (1 < c < M).

```

**% Variáveis de entrada:**

```

%      São fornecidas pela subrotina DADOS01 a qual é chamada antes do mapeamento.

```

**% Variáveis de saída:**

```

%      f      :Matriz com todos os valores do mapeamento.

```

**% Listagem comentada:**

```

% Cálculo dos valores iniciais B0 e C0
% Limpa algumas variáveis afim de evitar conflito de dimensões de matrizes

```

**limpa**

```

contador=4;
% chama a subrotina dados afim de carregá-los

```

**dados**

```

regiao=4
% Constrói-se dois vetores b e c dentro da região pré-estabelecida
maxb=10 * (Mdx - 1);
maxc=10 * (Mx - 1);
b=1.1:1:Md;
c=1.1:1:M;
for i=1:1:maxb,
    for j=1:1:maxc,
        ce(i,j)=c(j);
        be(i,j)=b(i);
    end
end
% Mapeia-se as funções {f, fi} com os vetores acima encontrados.
u=((1 - PD).^be).*((1 - p).^ce - 1));

```

```

ud=((1 - p).^ce).*((1 - PD).^(be - 1));
numr=p*ce.*u*(1 - g);
denr=numr + gf;
r=numr./denr;
f=h1*ce+h2*r - M;
fi=be.*ud*PD.*(1 - r)/sigd - Md;
clear be
clear ce
clear u
clear ud
clear numr
clear r
% Adota-se o valor de min=1 pois a área a ser mapeada é diferente de um
min=1;
% Chama a subrotina mat para calcular os valores de b0 e c0
mat
% O mapeamento em área unitária é feito subdividindo-se uma área qualquer
% num número inteiro de áreas unitárias e faz-se o mapeamento em cada
% sub-área separadamente.
% Faz-se o mapeamento em área unitária se:
% a - Foi escolhida a priori esta opção -> op03=1
% b - Durante o cálculo nesta área não unitária não foi encontrada raiz.
if op03=1
    detnull=length(fi4);
    if detnull=0
        % Faz-se o mapeamento por regiões menores com área unitária
        % Limpa algumas variáveis afim de evitar conflito de dimensões de matrizes
        limpa
        contador=4
        Constrói-se dois vetores b e c dentro da região pré-estabelecida
        clear kevalini
        clear kvalini
        clear ce
        clear be
        clear xvalini
        cvalini=1;
        While kevalini < Mx
            valini=kevalini + 1;
            c=kevalini + .01;
            vlini=1
            while kvalini < Mx;
                xvalini=kvalini + 1;
                ik=kvalini + .01;
                b=ik:.01:xvalini;
                c=ikc:.01:xcvalini;
                for i=1:1:99,
                    for j=1:1:99,
                        ce(i,j)=c(j);
                        be(i,j)=b(i);
                    end
                end
                kvalini=kvalini + 1;
                % Mapeia-se as funções {f, fi} com os vetores acima encontrados.
                u=((1 - PD).^be).*((1 - p).^ce - 1));
                ud=((1 - p).^ce).*((1 - PD).^(be - 1));
                numr=p*ce.*u*(1 - g);
                denr=numr + gf;
            end
        end
    end
end

```

```

        r=numr./denr;
        f=h1*ce+h2*r - M;
        fi=be.*ud*PD.*(1 - r)/sigd - Md;
        clear be
        clear ce
        clear u
        clear ud
        clear numr
        clear r
        % Chama a subrotina mat para calcular os valores de b0 e c0
    end
    min=0;
    mat
    kcvalini=kcvalini + 1;
end
end
end
end

```

### A1.25

**% Nome da subrotina: PROPPERD**

**% Descrição:**

% Esta subrotina é chamada por VALINI e calcula a probabilidade de perda de pacotes  
% de voz para sistemas com dados acoplados.

**% Variáveis de entrada:**

% Par ordenado (b,c) que é solução do sistema de equações.

**% Variáveis de saída:**

% Pdrp :Probabilidade de perda de pacotes e que pode ser um vetor bidimensional.  
% Par ordenado (b,c) que é solução do sistema de equações.

**% Listagem comentada:**

```

c=cx1;
b=bx1;
% como pode-se obter mais de uma solução há que se calcular todas soluções possíveis
compc=length(c)
for i=1:compc,
    uc
end
numx=ux.*c*p*(1 - g);
denx=numx + gf;
rx=numx./denx;
v1=(1 - g) .* (1 - (1 - rx). *p.*((1 - p).^c).*((1 - PD).^b));
% Cálculo da probabilidade de perda no equilíbrio
P1=(v1.^D) * gf* ((1 - gf)^(B - 1))./( 1 - v1.^N);
P2=gf*(1 - (1 - gf)*(v1.^2 * N));
P3=((1 - (1 - gf)*(v1.^N)).^2);
P4=(gf^2)*((1 - gf)^(B - 1))*(v1.^D - v1.^(B * N ));
Pdrop=P1.*(1 - P2./P3) + (gf^2)*(v1.^N)./P3 + P4;
format long
Pdrop;
Pdrp=Pdrop;
b;
c;

```

% Aqui termina o cálculo da probabilidade de perda de pacotes Pdrop. Neste ponto  
 % o aplicativo retorna a subrotina INI1.m

### A1.26

% Nome da subrotina: UC

#### % Descrição:

% Esta subrotina é chamada pela subrotinas RAIZ.m e PROPPERD.m calcula o valor de  
 %  $u(c)$ .

#### % Variáveis de entrada:

% b :Número de terminais de dados em disputa no equilíbrio.  
 % c :Número de terminais de voz em disputa no equilíbrio.  
 % PD :Probabilidade de acesso de terminais de dados no equilíbrio.  
 % c :Número de terminais de voz no equilíbrio.  
 % i :Valor que pode variar de um a três dependendo do número de soluções das  
 % raízes.

#### % Variáveis de saída:

% u :Vetor cujas dimensão pode ser 1, 2 ou 3 dependendo da dimensão de c.

#### % Listagem comentada:

```
if tiper=1
    if modelo=4
        if regiao=1
            ux(i)=(1 - PD).^b(i);
        elseif regiao=2
            ux(i)=((1 - p).^(c(i) - 1)).*((1 - PD).^b(i));
        elseif regiao=3
            ux(i)=((1 - PD).^b(i));
        elseif regiao=4
            ux(i)=((1 - PD).^b(i)).*((1 - p).^(c(i) - 1));
        end
        else
            if c < 1
                u=1;
            elseif c >=1
                u=(1 - p).^(c - 1);
            end
        end
    end
else
    if c < 1
        u=1;
    elseif c >=1
        u=(1 - p).^(c - 1);
    end
end
```

**A1.27****% Nome da subrotina: DADOS01****% Descrição:**

% Esta subrotina é chamada pela subrotina INI1 e tem a função de adequar  
 % os dados corretos para a opção escolhida dentre as várias oferecidas.

**% Variáveis de entrada:**

% As variáveis já estão disponíveis no programa a menos que o usuário deseje  
 % modificar.  
 % t1 :Duração média de um surto de voz.  
 % t2 :Duração média de um intervalo de silêncio.  
 % t3 :Duração média de um minisurto.  
 % t4 :Duração média de um minisilêncio.  
 % Rc :Taxa do canal de comunicação em bits/s.  
 % Rs :Taxa da fonte.  
 % H :Número de bits do cabeçalho.  
 % Dmax :Máximo atraso permitido para voz na transmissão.  
 % T :Duração do "Frame".  
 % M :Número de terminais de voz.

**% Variáveis de saída:**

% N :Número de pacotes por "Frame".  
 % tau :Duração de um "slot".  
 % g :Gama é a probabilidade que um surto de voz, com duração média t1 s  
 % termine num "slot" de duração média "tau" s.  
 % gf :Probabilidade que o surto termine no "frame" mais recente.  
 % gm :Probabilidade que um minisurto termine num "slot" de tempo.  
 % gmf :Probabilidade que um minisurto de voz termine no "frame" mais recente.  
 % sig :Probabilidade de que um intervalo de silêncio, de duração média t2 s,  
 % termine num "slot" de duração média tau s.  
 % sigm :Probabilidade de haver uma transição do SIL para TALK.  
 % alfa :Probabilidade de um minisurto ser o último no seu surto principal.  
 % B :Capacidade da memória (pacotes)  
 % D :Tempo máximo de espera em "slots" de tempo.

**% Listagem comentada:**

```
if constmod=0
    % O programa entra nesta primeira condicional se a opção de cálculo escolhida
    % foi para plotar a curva para vários pontos; só há necessidade de adequar os dados
    % na primeira vez que roda-se o programa pois para as demais os dados já
    % estão acertados.
    %-----Características do terminal de Voz-----
    t1=1.0;
    t2=1.35;
    t3=0.275;
    t4=0.050;
    Rc=720000;
    Rs=32000;
    H=64;
    Dmax=.032;
    if tiper=1
        T=0.016;
    end
    % Se o usuário escolheu modelo com dados acoplados, entra-se no IF.
    if modelodados=1
        if especMd=1
```

```

                Md=Mpj;
            else
                Md=M;
            end
            Rd=1200;
        end
    end
    % Chama-se a subrotina APRESENT.m para uma possível mudanças nos dados.
    % Deve-se criar uma variável de forma a evitar passar por esta instrução
    % várias vezes quando calcula-se a curva Pdrop=f(c).
    if multiponto=1
        apresent
        multiponto=2;
    end
    % Cálculos iniciais
    N=ceil(Rc * T/(Rs*T + H));
    tau=T/N;
    g=1 - exp(-tau/t1);
    gf=1 - (1 - g).^N;
    gm=1 - exp(-tau/t3);
    gmf=1 - (1 - gm) .^ N;
    sig=1 - exp(-tau/t2);
    sigm=1 - exp(-tau/t4);
    alfa=(t3 + t4)/t1;
    B=ceil(Dmax/T);
    D=ceil(Dmax/tau);
    if modelodados=1
        sigd=Rd/(Rs*N);
        h1=1 + g/sig;
        h2=N + gf/sig;
    end
    if modelodados=0
        if modelo=2
            raiz
        else
            % Chama-se a subrotina VE para calcular a probabilidade de um terminal
            % em disputa não conseguir reserva V(C,R).
            ve
        end
    elseif modelodados=2
        raiz
    end
end

```

### A1.28

% **Nome da subrotina: RAIZ**

% **Descrição:**

% Esta subrotina calcula os valores das raízes do sistema quando o modelo só  
 % inclui voz. Estas raízes podem ser simples ou triplas.

% **Variáveis de entrada:**

% Todas de DADOS01.m

% **Variáveis de saída:**

% c :Vetor raiz com solução simples ou três soluções.



```
%
      c=[c1 c2 c3]
%      Pdrop :Vetor probabilidade de perda com solução simples ou três soluções.
%      Pdrop=[Pdrop1 Pdrop2 Pdrop3]
```

**% Listagem comentada:**

% A constante w serve para desviar o cálculo de "c" quando "c" é única e maior que um.

```
w=0;
```

% Valor inicial do sinal de f(c), signf0, neste caso é sempre negativo

% pois para c=0, f(c)=-M.

```
signf0=-1;
```

% Varia-se c de zero até Cmax e calcula-se o sinal da função e

% observando se em dois pontos consecutivos ela muda de sinal.

```
Cmax=M;
```

```
for c=0:.01:Cmax;
```

% Modelo=1 cálculo com detector de atividade de voz rápido.

% Modelo=2 cálculo com detector de atividade de voz lento ponto equilíbrio.

% Modelo=3 cálculo com detector de atividade de voz lento.

% Modelo=4 cálculo com dados acoplados se tiper=1.

% Modelo=4 cálculo com detectores de voz rápidos pto equil se tiper=2.

**uc**

```
    if modelo=1
```

```
        K1=1 + ((1 - alfa) * gm / sigm) + (alfa * gm / sig);
```

```
        K2=N + ((1 - alfa) * gmf/sigm) + (alfa * gmf / sig);
```

```
        f=K1*c +K2*c*p*u(1 - gm)/(c*p*u(1 - gm) + gmf) - M;
```

```
    elseif modelo=2
```

```
        f=-M + c*(1 + g/sig) + c*p*u*(N + gf/sig)/(c*p*u + gf) ;
```

```
    elseif modelo=3
```

```
        f=-M + c*(1 + g/sig) + c*p*u*(N + gf/sig)/(c*p*u + gf) ;
```

```
    elseif modelo=4
```

```
        if tiper=2
```

```
            K1=1 + ((1 - alfa) * gm / sigm) + (alfa * gm / sig);
```

```
            K2=N + ((1 - alfa) * gmf/sigm) + (alfa * gmf / sig);
```

```
            f=K1*c +K2*c*p*u(1 - gm)/(c*p*u(1 - gm) + gmf) - M;
```

```
        end
```

```
    end
```

```
    if f < 0;
```

```
        signf=-1;
```

```
    else
```

```
        signf=+1;
```

```
    end
```

% Compara-se os sinais de F(c) do passo anterior e do atual

```
    if signf ~=signf0
```

% Adota-se um contador 'i' afim de criar um vetor das raízes c1, c2, c3

% o vetor chama-se - c - caso as raízes sejam múltiplas.

% Compara se houve mudanças nos sinais da função.

```
    signf0=signf;
```

% Calcula se a primeira raiz menor que um.

```
    if c < 1
```

```
        i=1;
```

% Chama-se a subrotina para calcular o valor da raiz.

```
        raiz1
```

```
        cr(i)=c;
```

```
        if modelo=2
```

```
            ve
```

```
        elseif modelo=4
```

```
            if tiper=2
```

```
                ve
```

```

                                end
                        else
                                tetr
                        end
% Pdrp(i) será um vetor de probabilidades de perdas.
Pdrp(i)=total;
i=i + 1;
w=1;
% O programa entra no elseif w=0 se a raiz for única e maior que um.
    elseif w=0
        i=1;
        raiz2
        cr(i)=c;
        if modelo=2
            ve
        elseif modelo=4
            if tiper=2
                ve
            end
        else
            tetr
        end
        Pdrp(i)=total(i);
        i=i + 1;
        w=w + 1;
% O programa entra neste desvio se já tiver calculado
% a primeira raiz e as duas outras sejam maiores que um.
    else
        raiz2
        cr(i)=c ;
        if modelo=2
            ve
        elseif modelo=4
            if tiper=2
                ve
            end
        else
            tetr
        end
        Pdrp(i)=total;
        i=i + 1;
    end
end
end
% Aqui tem-se o vetor raízes
c=cr;
% Aqui tem-se o vetor probabilidade de perda de pacotes relativos
% aos valores do vetor c dado acima.
Pdrop=Pdrp;

```

**A1.29****% Nome da subrotina: RAIZ1****% Descrição:**

% Esta subrotina é chamada por RAIZ.m e tem a função calcular as raízes da equação  
 %  $F(c) - M = 0$ , quando  $0 < c < 1$ , pelo método de Newton-Raphson.

**% Variáveis de entrada:**

% c :Valor inicial da raiz

**% Variáveis de saída:**

% cr :Raiz da função calculada pelo método de Newton-Raphson.

**% Listagem comentada:**

% Adota-se os seguintes valores iniciais

x=c;

k=1;

f1=100;

erro=0.0001;

% ----- Modelo completo -- -----

if modelo=1

**raiz0a**

%----- Modelo com detectores lentos pto equil -----

elseif modelo=2

**raiz1a**

% -----Modelo simplificado modificado -----

elseif modelo=3

**raiz1a**

% -----Vai para este desvio quando deseja-se calcular  $M=f(T)$

elseif modelo=4

    if tiper=2

**raiz0a**

    end

end

cr=c;

**A1.30****% Nome da subrotina: RAIZ0A****% Descrição:**

% Esta subrotina é chamada pela subrotina RAIZ1 e calcula a raiz da equação  
 %  $f(c)=M$  para o modelo com detectores rápidos.

**% Variáveis de entrada:**

% M :Número de terminais de voz.

% g :Gama é a probabilidade que um surto de voz, com duração média t1 s  
 % termine num "slot" de duração média "tau" s.

% gm :Probabilidade que um minisurto termine num "slot" de tempo.

% gmf :Probabilidade que um minisurto de voz termine no "frame" mais recente.

% K1 :calculado em RAIZ.

% K2 :Calculado em RAIZ.

% c :Valor inicial da raiz.

% p :Probabilidade de acesso dos terminais de voz.

**Variáveis de saída:**

```
%          c          :Raiz calculada
```

```
% Listagem comentada:
```

```
while k > erro;
    f=-M + K1*c + K2*c*p*(1 - gm)/(c*p*(1 - gm) + gmf);
    Df1=(c*K2*(p.^2)*((1 - gm).^2));
    Df2=((gmf + c*(1 - gm)*p).^2);
    Df3=(1 - gm)*K2*p/(gmf + c*(1 - gm)*p);
    DfDc=K1 - Df1/Df2 + Df3;
    cn=c - f/DfDc;
    k=abs(f1 - f);
    f1=f;
    c=cn;
end
```

### A1.31

```
% Nome da subrotina: RAIZ1A
```

```
% Descrição:
```

```
%          Esta subrotina calcula a primeira raiz quando o modelo do sistema utiliza
%          detectores lentos de voz. Esta probabilidade é calculada utilizando os dados
%          do ponto de equilíbrio ou por distribuição de probabilidades.
```

```
% Variáveis de entrada:
```

```
%          M          :Número de terminais de voz.
%          g          :Gama é a probabilidade que um surto de voz, com duração média t1 s
%          termine num "slot" de duração média "tau" s.
%          gm         :Probabilidade que um minisurto termine num "slot" de tempo.
%          gmf        :Probabilidade que um minisurto de voz termine no "frame" mais recente.
%          K1         :Calculado em RAIZ.
%          K2         :Calculado em RAIZ.
%          c          :Valor inicial da raiz.
%          p          :Probabilidade de acesso dos terminais de voz.
%          gf         :Probabilidade que um surto termine no "frame" mais recente.
%          sig        :Probabilidade que haja uma transição do estado falando para o de silêncio.
```

```
% Variáveis de saída:
```

```
%          c:          Raiz calculada.
```

```
% Listagem comentada:
```

```
while k > erro;
    h1=1 + g/sig;
    h2=N + gf/sig;
    err=c*p*(1-g)/(c*p*(1 - g) + gf);
    f=-M + h1*c + h2*err;
    Df1=c*(((1 - g)*p).^2)*h2;
    Df2=gf + c*(1 - g)*p;
    Df3=(1 - g)*p*h2;
    Df4=g/sig;
    DfDc=1 - Df1/(Df2.^2) + Df3/Df2 + Df4;
    cn=c - f/DfDc;
    k=abs(f1 - f);
    f1=f;
    c=cn;
    if c < 0
```

```

        c=x;
    end
end

```

### A1.32

#### % Nome da subrotina: VE

#### % Descrição:

```

%      Esta subrotina é chamada pela subrotina DADOS01 e RAIZ e calcula a probabilidade
%      de um terminal em disputa não conseguir reserva, dada pela expressão V(C,R).

```

#### % Variáveis de entrada:

```

%      M      :Número de terminais de voz.
%      g      :Gama é a probabilidade que um surto de voz, com duração média t1 s
%              termine num "slot" de duração média "tau" s.
%      gm     :Probabilidade que um minisurto termine num "slot" de tempo.
%      gmf    :Probabilidade que um minisurto de voz termine no "frame" mais recente.
%      K1     :Calculado em RAIZ.
%      K2     :calculado em RAIZ.
%      c      :Valor inicial da raiz.
%      p      :Probabilidade de acesso dos terminais de voz.
%      u      :Probabilidade incondicional.
%      gf     :Probabilidade que um surto termine no "frame" mais recente.

```

#### % Variáveis de saída:

```

%      vrc    :É uma matriz de dimensão (N X M) ver equação (3.23)

```

#### % Listagem comentada:

```

% Deve-se admitir um valor qualquer para vrc e depois limpá-lo para
% evitar conflito de dimensões quando roda-se o programa várias vezes.

```

```

vrc=1;
clear vrc
if modelo=2
% Cálculo de Pdrop utilizando o modelo com detectores lentos pelo ponto de equilíbrio.
    r=(c*p*u)*(1 - g)/(gf + c*p*u*(1 - g));
    vrc=(1 - g)*(1 - (1 - r)*p*((1 - p).^c));
elseif modelo=4
% Sistema com detectores rápidos utilizando os dados do ponto de equilíbrio.
    if tiper=2
% Cálculo da probabilidade de perda Pdrop para o desempenho em função de T.
        r=(c*p*u)*(1 - gm)/(gmf + c*p*u*(1 - gm));
        vrc=(1 - (1 - r)*p*((1 - p).^c));
    end
else
% Cria-se uma matriz V(R,C) de dimensão (M+1)XN.
    for i=1:M+1;
        C=i - 1;
        for j=1:N;
            R=j - 1;
            if C <=M-R;
                if modelo=1
% Cálculo de Pdrop utilizando detectores vocais rápidos.
                    vrc(i,j)=1 - (1 - gm)*(1 - R/N)*p*((1 - p).^C);
                elseif modelo=3

```

```

% Cálculo de Pdrop utilizando detectores vocais lentos.
                                vrc(i,j)=(1-g)*(1 - (1 - R/N)*p*((1 - p).^C));
                                end
                                else
                                vrc(i,j)=0;
                                end
                                end
                                end
                                end
end
% Chama-se a subrotina pdve afim de calcular Pdrop(V).
pdve

```

### A1.33

#### % Nome da subrotina: PDVE

#### % Descrição:

```

%      Esta subrotina é chamada por VE.m e calcula a probabilidade de perda de pacote;
%      probabilidade de um pacote em disputa não conseguir reserva dada por: P(ve).
%      Modelo=1 cálculo com detector rápido
%      Modelo=2 cálculo com detector lento Ponto de equilíbrio
%      Modelo=3 cálculo com detector lento
%      Modelo=4(*) cálculo com detector lento e com dados acoplados.
%      Modelo=4(**) cálculo com detector lento e com dados acoplados.
%      * Utilizado quando deseja-se encontrar Pdrop=f(M)
%      ** Utilizado quando deseja-se calcular a performance em função da
%      duração do "frame".

```

#### % Variáveis de entrada:

```

%      vr      :Matriz que corresponde a Ve(R,C) fornecida pela subrotina VE.m.
%      g      :Gama é a probabilidade que um surto de voz, com duração média t1 s
%              termine num "slot" de duração média "tau" s.
%      gm     :Probabilidade que um minisurto termine num "slot" de tempo.
%      gmf    :Probabilidade que um minisurto de voz termine no "frame" mais recente.
%      c      :Valor inicial da raiz.
%      p      :Probabilidade de acesso dos terminais de voz.
%      u      :Probabilidade incondicional.
%      gf     :Probabilidade que um surto termine no "frame" mais recente.

```

#### % Variáveis de saída:

```

%      pdv    :Probabilidade de perda de pacotes

```

#### % Listagem comentada:

```

if modelo=4
% Neste caso pode-se calcular Prop com dados acoplados -> tiper=1
% ou pode-se calcular Pdrop utilizando-se o modelo detectores rápidos; para levantar-se
% a curva M=F(T) -> tiper=2.
    if tiper=1
        p1=(vrc.^D)*gf*((1 - gf).^(B-1));
        p2=(1 - vrc.^N);
        p3=gf*(1-(1-gf)*(vrc.^(2*N)));
        p4=((1-(1-gf)*(vrc.^N)).^2);
        p5=(gf.^2)*(vrc.^N);
        p6=(gf.^2)*((1-gf).^(B-1))*((vrc.^D)-vrc.^(B*N));
        pdv=(p1./p2).*(1 - p3./p4) + p5./p4 + p6;
    end
end

```

```

elseif tiper=2
    p1=(vrc.^D)*gmf*((1 - gmf).^(B-1));
    p2=(1 - vrc.^N);
    p3=gmf*(1-(1-gmf)*(vrc.^(2*N)));
    p4=((1-(1-gmf)*(vrc.^N)).^2);
    p5=(gmf.^2)*(vrc.^N);
    p6=(gmf.^2)*((1-gmf).^(B-1))*((vrc.^D)-vrc.^(B*N));
    pdv=(p1./p2).*(1 - p3./p4) + p5./p4 + p6;
end
else
    if modelo=1
        p1=(vrc.^D)*gmf*((1 - gmf).^(B-1));
        p2=(1 - vrc.^N);
        p3=gmf*(1-(1-gmf)*(vrc.^(2*N)));
        p4=((1-(1-gmf)*(vrc.^N)).^2);
        p5=(gmf.^2)*(vrc.^N);
        p6=(gmf.^2)*((1-gmf).^(B-1))*((vrc.^D)-vrc.^(B*N));
        pdv=(p1./p2).*(1 - p3./p4) + p5./p4 + p6;
    elseif modelo=2
        pdv1
    elseif modelo=3
        pdv1
    end
end
clear p1
clear p2
clear p3
clear p4
clear p5
clear p6
if modelo=4
    drop
elseif modelo=2
    drop
else
    raiz
end

```

### A1.34

**% Nome da subrotina: PDV1**

**% Descrição:**

% Esta subrotina é chamada por PDVE.m e calcula a probabilidade de perda de pacotes pdv  
 % para um sistema utilizando modelo com detectores vocais lentos.

**% Variáveis de entrada:**

% vrc :Matriz de dimensão (N × M)  
 % g :Gama é a probabilidade que um surto de voz, com duração média t1 s  
 % termine num "slot" de duração média "tau" s.  
 % c :Valor inicial da raiz.  
 % p :Probabilidade de acesso dos terminais de voz.  
 % u :Probabilidade incondicional.  
 % gf :Probabilidade que um surto termine no "frame" mais recente.  
 % D :Atraso em unidades de "slots".

**% Variáveis de saída:**

%                    pdv        :Probabilidade de perda de pacotes.

**% Listagem comentada:**

```
p1=(vrc.^D)*gf*((1 - gf).^(B-1));
p2=(1 - vrc.^N);
p3=gf*(1-(1-gf)*(vrc.^(2*N)));
p4=((1-(1-gf)*(vrc.^N)).^2);
p5=(gf.^2)*(vrc.^N);
p6=(gf.^2)*((1-gf).^(B-1))*((vrc.^D)-vrc.^(B*N));
pdv=(p1./p2).*(1 - p3./p4) + p5./p4 + p6;
```

**A1.35****% Nome da subrotina: DROP****% Descrição:**

%                    Esta Subrotina é chamada por PDV.m ou TETCDR.m e calcula a probabilidade de perda  
 %                    de pacotes Pdrop do sistema de voz. Ela é chamada por TETCDR.m quando calcula  
 %                    Pdrop por densidade de probabilidades.

**% Variáveis de entrada:**

%                    pdv        :Matriz de dimensão (N X M) eq. (3.2)  
 %                    tetcdR    :Matriz de dimensão (N X M)  
 %                    TR        :Vetor de dimensão N.

**% Variáveis de saída:**

%                    Pdrp        :Probabilidade de perda de pacotes. Sistema com voz calculando a probabilidade  
 %                    de perda de pacotes pelo ponto de equilíbrio.

**% Listagem comentada:**

```
if modelo=2
    total=pdv;
elseif modelo=4
    if tiper=2
        total=pdv;
    end
else
    Pd1=pdv .* tetcdR;
    x=ones(1, M+1);
    pd2=(x * Pd1)';
    total=TR * pd2;
end
% Limpam-se algumas variáveis
clear Pd1
clear x
clear TR
clear tetcdR
if modelo=4
    Pdrop=total;
    Pdrp=total;
end
```



**A1.36****% Nome da subrotina: TETR****% Descrição:**

```
%
%      Esta subrotina é chamada por RAIZ.m e calcula a distribuição de probabilidade
%      dos "slots" estarem reservados que é denominada de TetR
%      Modelo=1 cálculo com modelo completo
%      Modelo=2 cálculo com detectores de atividade vocal lento pelo pto equi.
%      Modelo=3 cálculo com detectores de atividade vocal lento
%      Modelo=4(*) cálculo com modelo de voz com dados acoplados
%      Modelo=4(**) cálculo com detector rápido de voz ponto de equilíbrio
```

**% Variáveis de entrada:**

```
%      p      :Probabilidade de acesso dos terminais de voz.
%      u      :Probabilidade de acesso não contestada.
%      N      :Número de pacotes por "Frame".
%      gf     :Probabilidade que o surto termine no "frame" mais recente.
%      g      :Gama é a probabilidade que um surto de voz, com duração média t1 s.
%      r      :Probabilidade de um terminal estar reservado no equilíbrio.
%      R      :Número de terminais reservados.
%      tau    :Duração de um "slot".
%      gm     :Probabilidade que um minisurto termine num "slot" de tempo.
```

**% Variáveis de saída:**

```
%      TR     :Vetor de dimensão N (2.27).
```

**% Listagem comentada:**

```
if modelo=4
    compc=length(c);
    for i=1:compc,
        r(i)=c(i).*p.*u(i)*(1 - g)/(c(i).*p.*u(i)*(1 - g) + gf);
        for R=0:N - 1;
            er=R;
            comb;
            k=R + 1 ;
            vet(k)=k - 1;
            TR(i,k)=cm*(r(i).^R)*((1-r(i)).^(N-R));
        end
    end
else
    c=cr(i);
    uc
    % Detector de atividade vocal rápido
    if modelo=1
        r=c*p*u*(1 - gm)/(gmf + c*p*u*(1 - gm));
        for R=0:N - 1;
            er=R;
            comb;
            k=R + 1 ;
            vet(k)=k - 1;
            TR(k)=cm*(r.^R)*((1-r).^(N-R));
        end
        % Modelo simplificado com detector de atividade vocal lento
    elseif modelo=2
        r=c*p*u*(1 - g)/(c*p*u*(1 - g) + gf);
        for R=0:N - 1;
```

```

        er=R;
        comb;
        k=R + 1 ;
        vet(k)=k - 1;
        TR(k)=cm*(r.^R)*((1-r).^(N-R));
    end
%-----Modelo com detector de atividade vocal lento -----
elseif modelo=3
    r=c*p*u*(1 - g)/(c*p*u(1 - g) + gf);
    for R=0:N - 1;
        er=R;
        comb;
        k=R + 1 ;
        vet(k)=k - 1;
        TR(k)=cm*(r.^R)*((1-r).^(N-R));
    end
end
% -----Chamo a subrotina Tet(C/R) -----
tetrvpdv=pdv;
tecdv

```

### A1.37

% **Nome da subrotina: COMB**

% **Descrição:**

% Esta Subrotina é chamada por TETR.m e calcula a combinação de dois inteiros (N,R).

% **Variáveis de entrada:**

% N :Número de "slots" do "frame".

% R :Número de "slots" reservados.

% **Variáveis de saída:**

% cm :Combinação de [N,R].

% **Listagem comentada:**

```

for j=1:N-R ;
    x(j)=N+1-j;
    z=cumprod(x);
end
l=z(N-R);
for j=1:N-R;
    x(j)=N-R+1-j;
    z2=cumprod(x);
end
z3=z2(N-R);
cm=z1/z3;

```

**A1.38****% Nome da subrotina: TECDR****% Descrição:**

% Esta subrotina é chamada por TETR.m e calcula a expressão  $Tet(C/R)$  (2.28)

**% Variáveis de entrada:**

% c :Número de terminais em disputa no equilíbrio.  
 % p :Probabilidade de acesso dos terminais de voz.  
 % M :Número de terminais de voz do sistema.  
 % N :Número de pacotes por "Frame".  
 % R :Número de terminais reservados.  
 % C :Número de terminais em disputa.

**% Variáveis de saída:**

% TetcdR é uma matriz de dimensão  $N \times (M+1)$ .

**% Listagem comentada:**

```
clear tetcdR
po=1/(c + 1);
for k=1:M + 1;
    C=k - 1;
    for j=1:N;
        R=j - 1;
        if C < M-R;
            tetcdR(k,j)=po*((1 - po).^C);
        elseif C=M - R ;
            tetcdR(k,j)=(1 - po).^(M - R);
        else
            tetcdR(k,j)=0;
        end
    end
end
drop
```

**A1.39****% Nome da subrotina: RAIZ2****% Descrição:**

% Esta subrotina é chamada pela subrotina RAIZ e calcula as raízes da equação  $F(c)=M$   
 % quando  $c \geq 1$ , pelo método de Newton-Raphson.

**% Variáveis de entrada:**

% c: Valor inicial da raiz

**% Variáveis de saída:**

% cr: Raiz da função calculada pelo método de Newton-Raphson.

**% Listagem comentada:**

```
x=c;
f1=100;
erro=0.1;
% ----- Modelo Completo -----
if modelo=1
    raiz2a0
```

```

% ----- Modelo com detector lento de atividade vocal pto equi-----
elseif modelo=2
    raiz2a
%----- Modelo com detector lento de atividade vocal -----
elseif modelo=3
    aiz2a
elseif modelo=4
    if tiper=2
        raiz2a0
    end
end
% Se o algoritmo resultar num valor de c menor que zero, assume-se
% o valor inicial como solução.
if c < 0
    c=x;
end

```

#### A1.40

%Nome da subrotina: **RAIZ20A**

##### % Descrição:

```

%      Esta subrotina é chamada pela subrotina RAIZ2 e calcula a raiz da equação
%      f(c)=M para o modelo com detectores rápidos.

```

##### % Variáveis de entrada:

```

%      M      :Número de terminais de voz.
%      g      :Gama é a probabilidade que um surto de voz, com duração média t1 s
%              termine num "slot" de duração média "tau" s.
%      gm     :Probabilidade que um minisurto termine num "slot" de tempo.
%      gmf    :Probabilidade que um minisurto de voz termine no "frame" mais recente.
%      K1     :Calculado em RAIZ.
%      K2     :Calculado em RAIZ.
%      c      :Valor inicial da raiz.
%      p      :Probabilidade de acesso dos terminais de voz.

```

##### % Variáveis de saída:

```

%      c:      Raiz calculada

```

##### % Listagem comentada:

```

k=1;
while k > erro;
    u=(1 - p)^(c - 1);
    f=M + K1*c + K2*c*p*u*(1 - gm)/(c*p*u*(1 - gm) + gmf);
    Df1=c*(1 - gm)*K2*u*p*log10(1 - p);
    Df2=gmf + c*(1 - gm)*u*p;
    Df3=c*(1 - gm)*K2*u*p*((1 - gm)*u*p+c*(1 - gm)*u*p*log10(1 - p));
    fDc=K1 +Df1/Df2 - Df3/(Df2^2);
    cn=c - f/DfDc;
    k=abs(f1 - f);
    l=f;
    c=cn;
end

```

**A1.41****% Nome da subrotina: RAIZ2A****% Descrição:**

% Esta subrotina é chamada pela subrotina RAI2 e calcula a segunda raiz quando o  
 % modelo do sistema utiliza detectores lentos de voz esta probabilidade é calculada  
 % utilizando os dados do ponto de equilíbrio ou por distribuição de probabilidades.

**% Variáveis de entrada:**

% M :Número de terminais de voz.  
 % g :Gama é a probabilidade que um surto de voz, com duração média  $t_1$  s  
 % termine num "slot" de duração média "tau" s.  
 % gm :Probabilidade que um minisurto termine num "slot" de tempo.  
 % gmf :Probabilidade que um minisurto de voz termine no "frame" mais recente.  
 % K1 :Calculado em RAIZ.  
 % K2 :Calculado em RAIZ.  
 % c :Valor inicial da raiz.  
 % p :Probabilidade de acesso dos terminais de voz.  
 % gf :Probabilidade que um surto termine no "frame" mais recente.  
 % sig :Probabilidade que haja uma transição do estado falando para o de silêncio.

**% Variáveis de saída:**

% c: Raiz calculada

**% Listagem comentada:**

```
k=1;
while k > erro;
    u=(1 - p)^(c - 1);
    h1=1 + g/sig;
    h2=N + gf/sig ;
    f=- M + h1*c + h2*(c*p*u*(1 - g)/(c*p*u(1 - g)+ gf));
    Df1=(1 - g)*u*p*h2;
    Df2=gf + c*(1 - g)*u*p;
    Df3=c*(1 - g)*u*p*h2*log10(1 - p);
    Df4=c*(1 - g)*u*p*h2;
    Df5=(1 - g)*u*p + c*(1 - g)*u*p*log10(1 - p);
    DfDc=1 + Df1/Df2 + g/sig + Df3/Df2 + Df4*Df5/Df2.^2;
    cn=c - f/DfDc;
    k=abs(f1 - f);
    f1=f;
    c=cn;
end
```

**A1.42****% Nome da subrotina: ENTDURT****% Descrição:**

% Esta subrotina permite que o usuário escolha se deseja efetuar o cálculo apenas para um  
 % ponto ou vários. Na segunda opção, a subrotina verifica se  $T_{min} < T_{max}$ ; se por qualquer  
 % motivo isto não ocorrer, o aplicativo solicita ao usuário que digite novamente os valores.

**% Variáveis de entrada:**

% Op :Opção que pode assumir o valor 1 ou o valor 2 dependendo da opção do usuário.  
 % T :Valor de T(ms) quando o usuário deseja calcular apenas para um ponto.  
 % Tmin :Valor inicial de T(ms) quando deseja-se encontrar a curva  $M=f(T)$ .  
 % Tmax :Valor final de T(ms) quando deseja-se encontrar a curva  $M=f(T)$ .  
 % PassoT :Esta constante é o valor do incremento da variável cont (Subrot INI1).

**% Variáveis de saída:**

% Como esta subrotina só verifica as variáveis de entrada, as variáveis de  
 % saída são as mesmas de entrada.

**% Listagem comentada:**

```
Op=menu('Escolha a quantidade de pontos para o cálculo da Performance', ...
    '[Apenas um ponto]',...
    '[Vários pontos e curva M=f(T)]');
if Op=1
    disp('Entre com o da duração do frame em milisegundos.')
    T1=input('Duração do frame=');
    T=T1/1000;
    disp('Entre com o número de estações moveis - M.')
    M=input('Número de usuários=');
else
    disp('Entre com os valores, Tmin e Tmáx, duração do frame em milisegundos.')
    Tmin=input('Valor inicial=');
    Tmax=input('Valor final=');
    while Tmin > Tmax
        disp('Os valores estão incoerentes')
        disp('Entre novamente com os valores')
        Tmin=input('Valor inicial=');
        T1=Tmin;
        Tmax=input('Valor final=');
        T1=Tmax;
    end
    T1=Tmin;
    Tmin=T1/1000;
    T1=Tmax;
    Tmax=T1/1000;
    disp('Digite o valor do passo ')
    PassoT=input('PassoT=');
    passT=PassoT/1000;
    PassoT=passT;
end
```

**A1.43**

**% Nome da subrotina: APREPC**

**% Descrição:**

% Esta subrotina é chamada por INI1 ou por MULTRAIZ e apresenta o número de  
 % terminais em disputa no equilíbrio "c" e ou "b", a probabilidade de perda de pacotes  
 % Pdrop, o número de terminais do sistema M e a probabilidade de perda de pacotes p.  
 % Observa-se que b, c e Pdrop podem ser vetores ou matrizes dependendo se existem ou  
 % não raízes múltiplas para um dado valor de M.

**% Variáveis de entrada:**

% b, c, Pdrop, M e p.

**% Variáveis de saída:**

% As mesmas variáveis de entrada porém adequadamente organizadas.

**% Listagem comentada:**

```
if modelo=1
    disp('O sistema utiliza detectores de voz rápidos')
elseif modelo=2
    disp('O sistema utiliza detectores de voz lentos - [pto equilíbrio]')
elseif modelo=3
    disp('O sistema utiliza detectores de voz lentos')
elseif modelo=4
    disp('O sistema utiliza detectores de voz lentos com dados acoplados')
end
if modelo=4
    if Op=1
        % Apresenta-se os dados [M, p, b, c e Pdrop] pois a opção escolhida foi
        % para calcular apenas um ponto -> Op=1.
        M
        p
        format long
        b
        c
        Pdrop
    elseif Op=2
        % Cria-se o vetor zmat como sendo a soma das colunas da matriz c, onde c é
        % a matriz contendo a solução do número de terminais em disputa. Se houver
        % mais de uma solução para c, zmat terá dimensão 2X{(Mmáx-Mmin)/Passo}.
        zmat=cumsum(matrizc);
        for i=1:cont
            wvet(i)=2* matrizc(1,i);
            z1vet(i)=zmat(2,i);
            bvet(i)=matrizb(1, i)
            Pvet(i)=matrizP(1, i);
        end
        % wvet é um vetor cujas componentes tem o dobro do valor da primeira linha
        % da matriz z1vet; Pvet tem componentes iguais às primeiras linhas das
        % matrizes zmat e matrizP, respectivamente.
        vescal=abs(cumsum(z1vet - wvet));
        % Se vescal for igual a zero significa que todas as linhas da
        % matriz c são iguais, ou melhor, o sistema não tem múltiplas soluções.
        % Caso contrário deve-se apresentar as matrizes c e Pdrop.
        if vescal=0
            c=wvet/2
            b=bvet
        end
    end
end
```

```

        Pdrop=Pvet
    else
        for i=1:cont

            b1(i)=[matrizb( 1,i)];
            b2(i)=[matrizb( 2,i)];
            c1(i)=[matrizc( 1,i)];
            c2(i)=[matrizc( 2,i)];
            Pdrop1(i)=[matrizP(1,i)];
            Pdrop2(i)=[matrizP(2,i)];

        end
        vetM
        format long
        b=[b1; b2]
        c=[c1; c2]
        Pdrop=[Pdrop1; Pdrop2]
    end
end
else
    if Op=1
        M
        p
        format long
        c
        Pdrop
    elseif Op=2
        zmat=cumsum(matrizc);
        for i=1:cont
            wvet(i)=3* matrizc(1,i);
            z1vet(i)=zmat(3,i);
            Pvet(i)=matrizP(1, i);

        end
        vescal=abs(cumsum(z1vet - wvet));
        if vescal=0
            c=wvet/3
            Pdrop=Pvet
        else
            for i=1:cont
                c1(i)=[matrizc( 1,i)];
                c2(i)=[matrizc( 3,i)];
                Pdrop1(i)=[matrizP(1,i)];
                Pdrop2(i)=[matrizP(3,i)];

            end
            vetM
            format long
            c=[c1; c2]
            Pdrop=[Pdrop1; Pdrop2]
        end
    end
end
end
% O programa obtém o valor de ext e chama a subrotina extern afim de criar
% vários arquivos de acordo com a opção escolhida.
ext=menu('Escolha para que tipo de aplicativo você deseja disponibilizar os dados calculados', ...
        '[Plot-it]',...
        '[Word for windows]',...
        '[Plot-it e Word for Windows ]', ...
        '[Não utilizarei os dados em nenhum dos aplicativos acima]');

```



```

if ext=4
    boboplot=1;
else
    extern
end

```

#### A1.44

**% Nome da subrotina: EXTERN**

**% Descrição:**

```

%      Esta subrotina tem a função de colocar os dados calculados disponíveis para outros
%      aplicativos. Aqui os dados estão disponíveis para o Plot-it e/ou Word.

```

**% Variáveis de entrada:**

```

%      b, c Pdrop, M e p provenientes da subrotina APREPC.m

```

**% Variáveis de saída:**

```

%      São as mesmas variáveis de entradas porém preparadas adequadamente para
%      alguns aplicativos.

```

**% Listagem comentada:**

```

disp('Os dados disponíveis nesta versão são:')
disp('Uma matriz pdplot composta de duas colunas.')
disp('A primeira coluna é vetM que é um vetor com todos elementos de M')
disp('e a segunda coluna é Pdrop que é um vetor com a probabilidade de perda calculada')
disp('Obs. Os arquivos criados xxx.sdf e xxx.doc ficarão dentro do sub-diretório de trabalho ');
if vesca=0
    % Cria-se um arquivo para o aplicativo Plotit xxx.sdf ou Word xxx.doc
    pdplot=[vetM' Pdrop' bvet' c'];
    if ext=1
        save pplot.sdf pdplot -ascii
        disp('Os dados calculados já estão disponíveis para o Plot-it ');
    elseif ext=2
        save pplot.doc pdplot -ascii
        disp('Os dados calculados já estão disponíveis para o Word')
    elseif ext=3
        save pplot.sdf pdplot -ascii
        save pplot.doc pdplot -ascii
        disp('Os dados calculados já estão disponíveis para o Word e plot-it')
    end
else
    % Como tem-se duas soluções estáveis para Pdrop, cria-se dois arquivos pdplot1.sdf e
    % pdplot2.sdf para o aplicativo Plotit.
    pdplot1=[vetM' Pdrop1'];
    pdplot2=[vetM' Pdrop2'];
    if ext=1
        save pplot1.sdf pdplot1 -ascii
        save pplot2.sdf pdplot2 -ascii
        disp('Os dados calculados já estão disponíveis para o Plot-it')
    elseif ext=2
        save pplot1.doc pdplot1 -ascii
        save pplot2.doc pdplot2 -ascii
        disp('Os dados calculados já estão disponíveis para o Word')
    elseif ext=3
        save pplot1.sdf pdplot1 -ascii

```

```

        save pplot1.doc pdplot1 -ascii
        save pplot2.sdf pdplot2 -ascii
        save pplot2.doc pdplot2 -ascii
        disp('Os dados calculados já estão disponíveis para o Word e plot-it')
    end
end

```

#### A1.45

**% Nome da subrotina: APREPCT**

**% Descrição:**

% Esta subrotina é chamada por INI1 e apresenta os valores calculado para as  
% variáveis Pdrop e c, quando o usuário escolheu a opção performance.

**% Variáveis de entrada:**

% b, c, Pdrop, M e p.

**% Variáveis de saída:**

% As mesmas variáveis de entrada porém adequadamente organizadas.

**% Listagem comentada:**

```

if modelo=1
    disp('O sistema utiliza detectores de voz rápidos')
elseif modelo=2
    disp('O sistema utiliza detectores de voz lentos - [pto equilíbrio]')
elseif modelo=3
    disp('O sistema utiliza detectores de voz lentos')
elseif modelo=4
    disp('O sistema utiliza detectores de voz rápidos - [pto equilíbrio]')
end
if Op=1
    p
    c
    M
    T
    Pdrop
else
end
    ext=menu('Escolha para que tipo de aplicativo você deseja disponibilizar os dados calculados', ...
    '[Plot-it]',...
    '[Word for windows]',...
    '[Plot-it e Word for Windows ]', ...
    '[Não utilizarei os dados em nenhum dos aplicativos acima]');
if ext=4
    boboplot=1;
else
    extern
end

```

**A1.46**

**% Nome da subrotina: MULTRAIZ.**

**% Descrição:**

% Esta subrotina é chamada por INI1 e gera uma matriz Pdrop de dimensão  
 %  $[(M_{\text{max}} - m_{\text{min}}) \times 2]$  quando o sistema entra numa região de múltiplas raízes.

**% Variáveis de entrada:**

% b, c, Pdrop, M e p.

**% Variáveis de saída:**

% matrizb: Matriz com os valores calculados de b.  
 % matrizc: Matriz com os valores calculados de c.  
 % matrizP: Matriz com os valores calculados de Pdrop.

**% Listagem comentada:**

```
bob=length(c);
if bob=1
    cx=[c c c];
    if modelo=4
        bx=[b b b];
    end
    Pdx=[Pdrp Pdrp Pdrp];
else
    if modelo=4
        bx=b;
    end
    cx=c;
    Pdx=Pdrp;
end
if modelo=4
    for jteste=1:2
        matrizb(jteste,cont)=bx(1, jteste);
        matrizc(jteste,cont)=cx(1, jteste);
        matrizP(jteste,cont)=Pdx(1, jteste);
    end
else
    for jteste=1:3
        matrizc(jteste,cont)=cx(1, jteste);
        matrizP(jteste,cont)=Pdx(1, jteste);
    end
end
if M=Mmax
    aprepc
end
```

**A1.47**

**% Nome da subrotina: MC**

**% Descrição:**

% Esta subrotina é chamada por INI1 e calcula u(c) e M(c) para o modelo em estudo

**% Variáveis de entrada:**

% modelo: Tipo de cálculo utilizado.

% **Variáveis de saída:**

```
%
    teste      :Matriz com os dados que serão exportados para o Plot-it; estes dados
%              permitem traçar a curva de estabilidade do sistema  $F(c)=M$ .
```

% **Listagem comentada:**

```
if tiper=1
    if modelo=4
        k40 = 0; % constante bobb
    else
% ----- vetor u(c) -----
        for k=1:1:500
            xc=k/10;
            z(k)=xc;
            if xc < 1
                u(k)=1;
            elseif xc >=1
                u(k)=(1 - p).^(xc - 1);
            end
        end
% -----Modelo com detectores rápidos -----
        if modelo=1
            M1=u.*z*p*(1 - gm)*K2;
            M2=u.*z*p*(1 - gm) + gmf;
            M3=K1*z + M1./M2;
% -----Modelo com detectores de atividade de voz lentos -----
        else modelo=3
            h1=1 + g/sig;
            h2=N + gf/sig;
            M2=u.*z*p*(1 - g)./(u.*z*p*(1 - g)+ gf);
            M3=h1*z + h2*M2;
        end
% -----Plota F(c) qualquer para modelo -----
        figure(1)
        semilogx(z,M3)
        set(figure(1), 'numbertitle', 'on','menubar', 'none',...
            'name', 'FUNÇÃO DE EQUILÍBRIO DO SISTEMA', 'color', 'blue');
        xlabel('NÚMERO DE TERMINAIS EM DISPUTA');
        ylabel('F(c) ');
    end
% -----
    teste=[z' M3'];
    save teste.sdf teste -ascii
% -----Ploto Pdrop em função de c -----
    if Op=2
        if modelo=4
            figure(1)
            semilogy(vetM, Pdrop, 'y-')
            grid on
            set(figure(1), 'numbertitle', 'on','menubar', 'none',...
                'name', 'probabilidade de Perda', 'color', 'blue');
            xlabel('NÚMERO DE TERMINAIS EM CONVERSACÃO');
            ylabel('Pdrop ');
        else
            figure(2)
            semilogy(vetM, Pdrop, 'y-')
            grid on
            set(figure(2), 'numbertitle', 'on','menubar', 'none',...
```

```

        'name', 'probabilidade de Perda', 'color', 'blue');
        xlabel('NÚMERO DE TERMINAIS EM CONVERSAÇÃO');
        ylabel('Pdrop ');
    end
end
else
    figure(2)
    plot(vetT, vetM, 'y-')
    grid on
    set(gcf, 'numbertitle', 'on', 'menubar', 'none', ...
        'name', 'T x Pdrop(1%)', 'color', 'blue');
    xlabel('Duração do frame');
    ylabel('M ');
end
end

```

#### A1.48

**% Nome da subrotina: APRESENT**

**% Descrição:**

% Esta subrotina é chamada pela subrotina DADOS01, mostra os valores das variáveis  
 % e tem como opção a modificação de alguns dados caso o usuário deseje.

**% Variáveis de entrada:**

% Além de todas da subrotina DADOS01 acrescenta-se a variável modifc que pode assumir  
 % o valor 1 quando usuário não deseja modificar os dados existentes ou o valor 2 quando  
 % o usuário deseja efetuar modificação nos dados existentes.

**% Variáveis de saída:**

% São as mesmas da subrotina DADOS01 e alguns valores poderão ter sido alterados se o  
 % usuário desejar.

**% Listagem comentada:**

```

% Cria-se a constante Operrad afim de não passar duas vezes por esta opção quando o aplicativo
% está levantando dados para serem alocados numa curva do tipo Pdrop=f(M).
Operrad=0;
while Operrad=0
    disp(' Abaixo segue os valores atuais das variáveis do problema ');
    p
    % O aplicativo entra na primeira condicional se não existir dados acoplados.
    if modelodados=0
        % O aplicativo entra na segunda condicional se o sistema estiver calculando a curva
        M=f(c).
        if Op=1
            M
        else
            if tiper=1
                Mmin
                Mmax
            else
                Tmin
                Tmax
            end
        end
    end
end
end
Rc

```

```

Rs
T
Dmax
if modelodados=1
    if especMd=1
        Md=Mpj
    else
        Md=M
    end
    M
    Rd=1200
end
modific=menu('Após verificar os dados apresentados escolha:', ...
    '[Não desejo modificar os dados]',...
    '[Farei modificações]');
if modific=2
    modif
    Operrad=1;
else
    disp('O programa vai utilizar os dados acima nos cálculos');
    Operrad=1;
end
end
end

```

#### A1.49

**% Nome da subrotina: MODIF**

**% Descrição:**

% Esta subrotina é chamada pela subrotina APRESENT e permite ao usuário modificar os  
 % parâmetros do sistema PRMA.

**% Variáveis de entrada:**

% Escolha :pode assumir cinco valores dependendo da variável a ser modificada ou se o  
 % usuário desejar sair desta subrotina.

**% Variáveis de saída:**

% São as mesmas da subrotina DADOS01 mas alguns valores poderão ter sido alterados.

**% Listagem comentada:**

```

variável=0
while variavel=0
    escolha=menu('Escolha a opção à modificar', ...
        '[Prob de Acesso - p]',...
        '[Número de Terminais - M ]','[Taxa do Canal - Rc ]', ...
        '[Taxa da Fonte - Rs]','[Duração do Frame - T]',...
        '[Tempo Máximo de Espera - Dmáx]',...
        '[Sair]' );
    if escolha=1
        variavel=0;
        testprob
    elseif escolha=2
        tespon
    elseif escolha=3
        variável=0;
        Rc=input('Rc=');
    end
end

```

```

elseif escolha=4
    variável=0;
    Rs=input('Rs=');
elseif escolha=5
    variável=0;
    T1=input('Entre com o valor da duração do frame em (ms)=');
    T=.001*T1
elseif escolha=6
    variável=0;
    Dmax=input('Dmax=');
else
    variável=1;
end
end
% Cria-se esta constante para evitar que o programa utilize os valores padrões.
constmod=1;

```

### A1.50

#### % Nome da subrotina: ATRASO

#### % Descrição:

Esta subrotina calcula o atraso de um sistema PRMA com dados acoplados

#### % Variáveis de entrada:

```

%      M      :Vetor com os valor do número de terminais do sistema
%      c      :Vetor com o número de terminais de voz em disputa no equilíbrio
%      b      :Vetor com o número de terminais de dados em disputa no equilíbrio
%      p      :Probabilidade de acesso dos terminais de voz
%      pd     :Probabilidade de acesso dos terminais de dados
%      beta   :Parâmetro de sistema
%      RS     :Taxa de transmissão da fonte
%      Rc     :Taxa de transmissão do canal
%      T      :Duração do "frame"

```

#### % Variáveis de saída:

```

%      espmed :Tempo médio de espera
%      espbeta :Tempo de espera para 95% dos terminais.

```

#### % Listagem comentada:

```

% Entrada dos dados:
M = [22 24 26 28 30 32 34 36];
c = [.0508 .0612 .0739 .0899 .1005 .1379 .1791 .2476];
b = [ .3945 0.4691 0.5585 0.6674 0.8028 0.9754 1.2811 1.8218 ];
p = .3;
PD = .2;
beta = 0.005;
RS = 32000;
Rc = 720000;
T = .016;
% Início dos cálculos:
N = ceil(Rc * T/(Rs*T + H));
sigd = Rd/Rs*N;
Md = M;
rod = b./Md;
% calcvar encontra o valor de u e ud para os vários valores de M, c e b

```

```

% dentro das vária regiões.
calcvar
numr = p*c.*u*(1 - g);
denr = numr + gf;
r = numr./denr;
w = PD*ud.*(1 - r);
espmed = 1./(w.*(1 - rod));
espbeta = (1./w).*(1 + log(beta)./log(rod));
% Prepara-se a figura
figure(1)
    semilogy(M,espmed,M, espbeta)
    set(figure(1), 'numbertitle', 'on','menubar', 'none',...
        'name', 'ATRASSO DOS PACOTES DE DADOS', 'color', 'blue');
    xlabel('NÚMERO DE TERMINAIS');
    ylabel('ATRASSO - ms ');
    end
med = [M' espmed' espbeta'];
save med.sdf med -ascii

```

### A1.51

**% Nome da subrotina: CALCVAR**

**% Descrição:**

Esta subrotina é chamada por ATRASSO.m e encontra o valor de u e ud para as várias regiões.

**% Variáveis de entrada:**

```

%           M       :Vetor com os valor do número de terminais do sistema
%           c       :Vetor com o número de terminais de voz em disputa no equilíbrio
%           b       :Vetor com o número de terminais de dados em disputa no equilíbrio
%           p       :Probabilidade de acesso dos terminais de voz
%           pd      :Probabilidade de acesso dos terminais de dados

```

**% Variáveis de saída:**

```

%           u       :Probabilidade de acesso não contestada dos terminais de voz
%           ud      :Probabilidade de acesso não contestada dos terminais de dados

```

**% Listagem comentada:**

```

compM = length(M);
for i = 1:compM
    if c(i) < 1
        if b(i) < 1
            % reg01
            u(i) = (1 - PD).^b(i);
            ud(i) = (1 - p).^c(i);
        end
    end
    if c(i) > 1
        if b(i) < 1
            % reg02
            u(i) = ((1 - p).^(c(i) - 1)).*((1 - PD).^b(i));
            ud(i) = (1 - p).^c(i);
        end
    end
end

```



```

        if c(i) < 1
            if b(i) > 1
                % reg03
                u(i) = ((1 - PD).^b(i));
                ud(i) = ((1 - p).^c(i)).*((1 - PD).^b(i) - 1));
            end
        end
        if c(i) > 1
            if b(i) > 1
                %reg04
                u(i) = ((1 - PD).^b(i)).*((1 - p).^c(i) - 1));
                ud(i) = ((1 - p).^c(i)).*((1 - PD).^b(i) - 1));
            end
        end
    end
end

```

### A1.52

**% Nome da subrotina: Limpa**

**% Descrição:**

% Limpar algumas variáveis afim de evitar conflito de dimensão entre variáveis.

**% Listagem:**

```

clear abf
clear abfi
clear A11
clear A12
clear A22
clear A23
clear b
clear be
clear c
clear ce
clear const
clear denr
clear dima
clear dimb
clear f
clear fi
clear i
clear j
clear k
clear l
clear numr
clear Pe
clear r
clear res
clear u
clear x0
clear x11
clear x22
clear y0
clear z
clear pdv
clear Pd1

```

```
clear x  
clear TR  
clear tetcdR
```

## ANEXO 2

### ESTABILIDADE DO PRMA

Seja  $G(c)$ , dado pela equação (A2.01) abaixo, o fluxo de saída do estado CON correspondente ao modelo de sistema da figura A-01.

Ao estudar-se a estabilidade do sistema PRMA no ponto de equilíbrio, verifica-se que esta depende do sinal da derivada  $G'(c)$ . Para um sistema estável a medida que  $c$  aumenta (diminui) o mesmo deve ocorrer com  $G(c)$ .

$$G(c) = c\gamma_m + \text{cpu}(1 - \gamma_m)(1 - r) \quad (\text{A2.01})$$

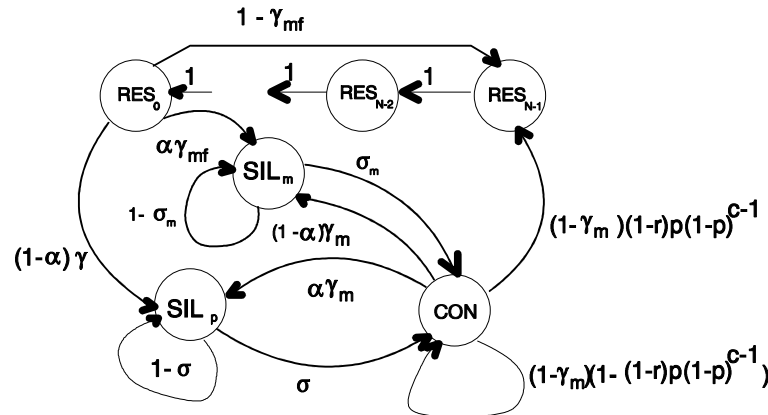


fig. A-01

Com o aumento do número de terminais em disputa, o fluxo de saída deste estado CON deverá aumentar, logo a derivada de  $G(c)$  deverá também ser positiva e desta forma não haverá congestionamento do sistema. Verificar-se-á que esta derivada relaciona-se com a derivada de  $F(c)$  através da relação  $F'(c) > 0 \Rightarrow G'(c) > 0$ .

Do estado RES pode-se escrever a equação:

$$r(1 - \gamma_{mf}) + \text{cpu}(1 - r)(1 - \gamma_m) - r = 0 \quad (\text{A2.02})$$

e desta pode-se reescrever  $G(c)$  na forma:

$$G(c) = c\gamma_m + r - r(1 - \gamma_{mf}) \quad (A2.03)$$

Por outro lado, a função de equilíbrio do sistema  $F(c)$  é dada por (A2.04)

$$F(c) = c \left[ 1 + \frac{(1-\alpha)\gamma_m}{\sigma_N} + \frac{\alpha\gamma_m}{\sigma} \right] + \frac{cpu(1-\gamma_m)}{cpu(1-\gamma_m) + \gamma_{mf}} \left[ N + \frac{(1-\alpha)\gamma_{mf}}{\sigma_N} + \frac{\alpha\gamma_{mf}}{\sigma} \right] \quad (A2.04)$$

Derivando  $G(c)$  e  $F(c)$  em relação a  $c$  obtém-se:

$$G'(c) = \gamma_m + r'(c)\gamma_{mf} \quad (A2.05)$$

$$F'(c) = A + Br'(c) \quad (A2.06)$$

com  $A$  e  $B$  dados por:

$$A = 1 + \frac{\gamma_m(1-\alpha)}{\sigma_m} + \frac{\gamma_m\alpha}{\sigma} \quad B = N + \frac{\gamma_{mf}(1-\alpha)}{\sigma_m} + \frac{\gamma_{mf}\alpha}{\sigma} \quad (A2.07)$$

Como as constantes  $A$  e  $B$  são sempre positivas, se  $r'(c) > 0 \Rightarrow G'(c) > 0$  e  $F'(c) > 0$

Simplificando as equações pode-se escrever:

$$\gamma_m F'(c) - AG'(c) = r'(c)[N\gamma_m - \gamma_{mf}] \quad (A2.08)$$

Quando  $\gamma_{mf} \approx N\gamma_m$  o lado direito é próximo de zero e como  $A > 0$ , o sinal de  $F'(c)$  e  $G'(c)$  é o mesmo para a maioria dos valores de  $c$ . Considerando esta hipótese com mais detalhes:

Como  $\gamma_{mf} = [1 - (1 - \gamma_m)^N]$  pode-se escrever:

$$N\gamma_m - \gamma_{mf} \approx \frac{N[N-1]}{2} \gamma_m^2 > 0 \quad (A2.09)$$

Se  $r'(c) < 0$  então  $AG'(c) > \gamma_{mf}F'(c)$  e portanto se  $F'(c) > 0 \Rightarrow G'(c) > 0$ .

Analisando-se o sinal das derivadas para valores de  $c$  menores e maiores que um, de (A2.02):

$$r(c) = \frac{cpu(c)[1 - \gamma_m]}{cpu(c)(1 - \gamma_m) + \gamma_{mf}} \quad (A2.10)$$

onde:

$$u(c) = \begin{cases} (1-p)^{c-1} & \text{para } c \geq 1 \\ 1 & \text{para } c < 1 \end{cases} \quad (A2.11)$$

Em primeiro lugar considera-se  $c < 1$ . Calculando  $\frac{d'r(c)}{dc}$  chega-se a:

$$\frac{dr(c)}{dc} = \frac{r - r^2}{c} = \frac{r(1-r)}{c} \quad (A2.12)$$

De (A2.12), tem-se que  $r'(c) > 0$ .

Considera-se agora  $c \geq 1$ . Neste caso como  $u(c) = (1-p)^{c-1}$  tem-se:

$$\frac{du(c)}{dc} = (1-p)^{c-1} \ln(1-p) \quad (A2.13)$$

e destas com (A2.10) obtém-se:

$$\frac{dr(c)}{dc} = r(c) \left\{ 1 - r(c) \right\} \left( \frac{1}{c} + \ln(1-p) \right) \quad (A2.14)$$

Novamente como  $0 < r < 1$ ,  $r'(c)=0$  somente em  $c=c_0$ , onde  $c_0$  é a solução de:

$$\frac{1}{c_0} + \ln(1-p) = 0 \Rightarrow c_0 = \frac{-1}{\ln(1-p)} \quad (A2.15)$$

para  $c > c_0$  a função  $r'(c)$  é negativa. Define-se o sistema como eficiente quando  $c < c_0$ , e neste caso o "throughput"  $r$ , aumenta com o aumento do tráfego oferecido, isto é, quando  $r'(c) > 0$ . Sendo  $r'(c) > 0$   $G'(c)$  e  $F'(c)$  têm o mesmo sinal.

Desta forma para todos valores de  $c$ , quando  $F'(c) > 0 \Rightarrow G'(c) > 0$ . A recíproca não é sempre verdadeira pois é possível que ocorram casos em que  $G'(c) > 0$  enquanto  $F'(c) < 0$ . Isto ocorre para uma pequena fração de valores de  $c$  onde ambos  $F'(c)$  e  $G'(c)$  são próximos a zero. Neste conjunto de valores de  $c$ ,  $r'(c) < 0$  e  $G'(c)$  é pequeno e positivo, de tal forma que

$AG'(c) + r'(c)[N\gamma_m - \gamma_{mf}] < 0$ . Verifica-se assim que  $F'(c) < 0$  para pontos próximo ao mínimo local ou ao máximo local de  $F(c)$  e pode ser estável. Logo a estabilidade, neste caso, deve ser verificada calculando-se  $G'(c)$ .

## **ANEXO 3**

### **ESTATÍSTICA DA PERDA DE PACOTES**

Neste anexo estuda-se a estatística de perda de pacotes para um sistema com detectores de atividade de voz lentos. Nesta análise, deve-se assumir que o sistema **PRMA** está no equilíbrio e assim num instante particular o número de terminais em disputa é **c** e o número de terminais em reserva nos próximos **N** "slots" (um "frame") é **R**. Para calcular a probabilidade de perda de pacotes considere-se um terminal com um surto, com **L** pacotes, começando no "slot" atual. Deve-se encontrar probabilidade de que **k** ( $k=0, 1, \dots, L$ ) pacotes sejam perdidos antes que o terminal obtenha reserva.

#### 1 - Análise Básica

Um terminal que iniciou um novo surto de voz deve disputar com **C** terminais já no estado **CON** e para obter uma reserva, são necessárias três condições:

- a - O slot não está reservado com probabilidade  $(1 - R/N)$ ;
- b - O terminal tem permissão para transmitir com probabilidade **p**;
- c - Os outros **C** terminais em disputa não tem permissão com probabilidade  $(1 - p)^C$

O terminal desta forma fica no estado **CON**, estado de disputa, até que todas as três condições sejam satisfeitas simultaneamente. Denota-se esta probabilidade por **v(C,R)**. Os três eventos considerados acima são independentes. Dada uma certa probabilidade de que um "slot" está reservado, a condição de reserva de um dado slot é independente dos eventos de permissão gerados em um terminal em disputa e isto nos leva a expressão abaixo.

$$v(C,R) = 1 - \left(1 - \frac{R}{N}\right) p (1 - p)^C \quad (A3.01)$$

A técnica de descarte é utilizada aqui de tal modo que o terminal descarta pacotes de voz que foram armazenados por mais tempo (fixado em  $\mathbf{D}$  "slots"). Assume-se que  $\mathbf{D}$  é um número inteiro múltiplo de  $N$ . O número de pacotes perdidos no início de um surto de voz depende de quanto tempo o terminal espera por uma reserva. Desenvolve-se a estatística de perda de pacotes como uma função da variável aleatória  $v$ , definida acima. Observe-se que no início de um surto, um terminal espera  $j$  slots para obter uma reserva com uma probabilidade  $P_w(j)$  e esta probabilidade tem uma distribuição geométrica.

$$P_w(j) = (1 - v)v^{j-1}, j = 1, 2, 3, \dots \quad (\text{A3-02})$$

Num surto de comprimento  $L$  pacotes, nenhum pacote é perdido se  $j \leq \mathbf{D}$  slots, onde  $\mathbf{D}$  é o limite de atraso na transmissão de voz. Após esperar  $\mathbf{D}$  slots, o terminal perde o pacote inicial do surto. O segundo pacote esperou certamente  $N$  slots menos que o primeiro. Desta forma, o terminal perde o segundo pacote se ele ainda está disputando uma reserva após  $\mathbf{D} + N$  "slots". Se após  $\mathbf{D} + (L - 1)N$  "slots" de tempo, o terminal ainda não conseguiu reserva, ele perde todo o surto composto pelos  $L$  pacotes.

A figura (A3-01) evidencia  $\eta_{\text{drop}}(j)$ , o número de pacotes perdidos como uma função do tempo de espera.

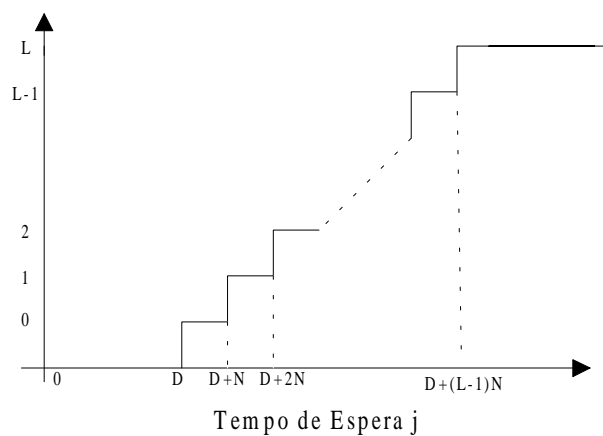


fig. A3-01



No cálculo do número de pacotes perdidos para os diversos valores de  $j$  tem-se:

$$\begin{cases} \eta_{\text{perda}}(j) = 0, & \text{se } 1 \leq j \leq D \\ \eta_{\text{perda}}(j) = K, & \text{se } D + (K-1)N \leq j \leq D + KN \\ \eta_{\text{perda}}(j) = L, & \text{se } D + (L-1)N + 1 \leq j \end{cases} \quad (\text{A3.03})$$

De (A3.02) e (A3.03), demonstra-se na ref.[2] que  $P_{\text{drop}}(\eta_{\text{perda}} = K/L)$ , que representa a probabilidade que num surto de  $L$  pacotes  $K$  sejam perdidos, é dada por:

$$P_r(\eta_{\text{drop}} = 0|L) = \sum_{j=1}^D P_w(j) = 1 - v^D \quad (\text{A3.04})$$

$$P_r(\eta_{\text{drop}} = K|L) = \sum_{j=D+(K-1)N+1}^{D+KN} P_w(j) = v^{D+(K-1)N+1} - v^{D+KN} \quad (\text{A3.05})$$

$$P_r(\eta_{\text{drop}} = L|L) = \sum_{j=D+(L-1)N+1}^{\infty} P_w(j) = v^{D+(L-1)N} \quad (\text{A3.06})$$

Para um surto com  $L$  pacotes, o número médio de pacotes perdidos é:

$$E[\eta_{\text{drop}} | L] = \sum_{K=0}^L K P_r(\eta_{\text{drop}} = \frac{K}{L}) \quad (\text{A3.07})$$

que simplificado resulta em:

$$E[\eta_{\text{drop}} | L] = v^D \frac{1 - v^{NL}}{1 - v^N} \quad (\text{A3.08})$$

Desta forma considerando-se todos os surtos de voz, o número médio de pacotes perdidos é a média de  $E[\eta_{\text{drop}} | L]$  com respeito à probabilidade de existirem  $L$  pacotes no surto. Esta probabilidade é dada por:

$$P_r(L) = \gamma_f (1 - \gamma_f)^{L-1} \quad (\text{A3.09})$$

Onde  $\gamma_f$  é a probabilidade que o surto termine no frame mais recente (isto é, que saia do estado RES e vá para o estado SIL). Assim,  $(1 - \gamma_f)$  é a probabilidade de permanecer no estado RES e  $(1 - \gamma_f)^{L-1}$  a de que existam  $L-1$  pacotes a serem transmitidos (que podem ser perdidos).

De (A3-09) pode-se escrever:

$$E[\eta_{\text{drop}}] = \sum_{L=1}^{\infty} E[\eta_{\text{drop}} | L] P_r(L) \quad (\text{A3.10})$$

e após simplificar-se resulta:

$$E[\eta_{\text{drop}}] = \frac{v^D}{1 - v^N (1 - \gamma_f)} \quad (\text{A3.11})$$

Observe que não se está restringindo  $D$  como sendo um múltiplo inteiro de  $N$ . Um surto de  $L$  pacotes dura  $LN$  "slots".

A probabilidade de perda de pacotes  $P_{\text{drop}}$  pode ser escrita como sendo a razão entre o número médio de pacotes perdidos e o inverso da probabilidade de transição de "frame".

$$P_{\text{drop}} = \frac{v^D}{1 - v^N (1 - \gamma_f)} \gamma_f \quad (\text{A3.12})$$

## ANEXO 4

### DESCRIÇÃO DO MÉTODO DE NEWTON RAPHSON (NR) NA RESOLUÇÃO DE UM SISTEMA NÃO LINEAR

Neste anexo descreve-se o método de NR para a resolução de um sistema de equações não lineares, resultante do sistema PRMA com dados acoplados.

#### **I.1 - Modelos adotados**

Ao estudar-se sistemas PRMA foi adotado o modelo de detectores de atividade de voz lento e com dados acoplados. O diagrama de estados para voz e dados acoplados correspondente é o do fig. A4.01.

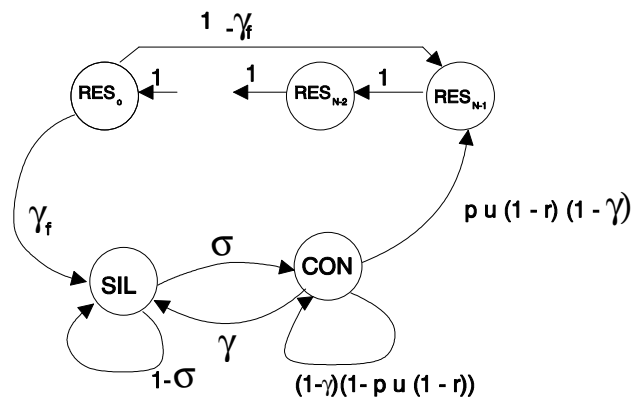


fig. A4.01

e para cada terminal de dados foi adotado o processo de vida-morte representado na fig. A4.02.

$$x = \sigma d w + (1 - w) (1 - \sigma d)$$

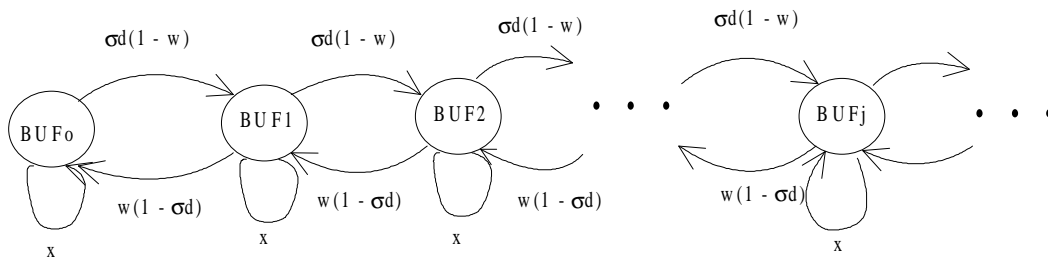


fig. A4.02

## I.2 - Obtenção do sistema de equações

Analisando-se o ponto de equilíbrio do sistema, tem-se o vetor  $\{s, c, r_i, b_j, 0 \leq i \leq N-1, j \geq 0\}$ , onde os valores das variáveis pertencentes ao conjunto correspondente ao ponto de equilíbrio são escritos em letras minúsculas.

$c = \text{eq}\{C\} =$  número de terminais em equilíbrio no estado CON

$s = \text{eq}\{S\} =$  número de terminais em equilíbrio no estado SIL

$r_i = \text{eq}\{R_i\} =$  número de terminais em equilíbrio no estado  $\text{RES}_i$

$b_j = \text{eq}\{\text{BUF}_j\} =$  número de terminais de dados em equilíbrio no estado  $\text{BUF}_j$

Denotando-se as probabilidades incondicionais para voz e dados como sendo  $u(b,c)$  e  $u_d(b,c)$  tem-se:

$$u_d = f1(b,c) \quad (\text{A4.01})$$

$$u = g1(b,c) \quad (\text{A4.02})$$

De acordo com o diagrama da fig. A4.01, no equilíbrio, pode-se escrever para cada estado:

Estado RES no Buffer N-1:

$$r(1 - \gamma_f) + \text{cpu}(1 - r)(1 - \gamma) - r = 0 \quad (\text{A4.03})$$

Estado SIL:

$$s\sigma = r\gamma_f + c\gamma \quad (\text{A4.04})$$

e como o número total de terminais de voz é M tem-se:

$$s + c + Nr = M \quad (\text{A4.05})$$

Obtém-se assim um sistema com três equações a três incógnitas  $c$ ,  $s$  e  $r$ .

$$\begin{cases} r(1-\gamma_f) + \text{cpu}(1-r)(1-\gamma) - r = 0 \\ r\gamma_f + \gamma c - s\sigma = 0 \\ s + c + Nr - M = 0 \end{cases} \quad (\text{A4.06})$$

Simplificando-se chega-se ao seguinte sistema de equações a duas incógnitas:

$$\begin{cases} c\left(1 + \frac{\gamma}{\sigma}\right) + r\left(M + \frac{\gamma_f}{\sigma}\right) - M = 0 \\ \frac{b u_d p_d (1-r)}{\sigma_d} - M_d = 0 \end{cases} \quad (\text{A4.07})$$

Com  $r = \frac{\text{cpu}(1-\gamma)}{\text{cpu}(1-\gamma) + \gamma_f}$

### I.3 - Resolução do sistema de equações

Neste item descreve-se o método de resolução do sistema (A4.07) que pode ser reescrito na forma:

$$\begin{cases} f(b,c) = h_1 \cdot c + h_2 \cdot r - M = 0 \\ \phi(b,c) = \frac{b u_d k_d (1-r)}{\sigma_d} - M_d = 0 \end{cases} \quad (\text{A4.08})$$

com:  $h_1 = \left(1 + \frac{\gamma}{\sigma}\right)$  e  $h_2 = \left(M + \frac{\gamma_f}{\sigma}\right)$

Calculando-se as derivadas parciais de  $f$  e  $\phi$  em relação a  $b$  e  $c$  encontram-se:

$$\left\{ \begin{array}{l} \frac{\partial \varphi}{\partial b} = b \cdot k_d \frac{\partial u d}{\partial b} - k_d \cdot b \cdot \left( r \cdot \frac{\partial u d}{\partial b} + u d \cdot \frac{\partial r}{\partial b} \right) + k_d \cdot u d - k_d \cdot u d \cdot r \end{array} \right. \quad (A4.09)$$

$$\left\{ \begin{array}{l} \frac{\partial \varphi}{\partial c} = b \cdot k_d \frac{\partial u d}{\partial c} - k_d \cdot b \cdot \left( r \cdot \frac{\partial u d}{\partial c} + u d \frac{\partial r}{\partial c} \right) \end{array} \right. \quad (A4.10)$$

$$\frac{\partial f}{\partial b} = h_2 \cdot \frac{\partial r}{\partial b} \quad (A4.11)$$

$$\frac{\partial f}{\partial c} = h_2 \cdot \frac{\partial r}{\partial c} \quad (A4.12)$$

Como  $h_1$ ,  $h_2$ ,  $\sigma_d$ ,  $M$  e  $M_d$  são constantes, de uma forma geral, não levando-se em consideração a região a ser pesquisada, pode-se escrever:

$$\left\{ \begin{array}{l} b = \xi = b_0 + \Delta b \\ c = \eta = c_0 + \Delta c \end{array} \right. \quad (A4.13)$$

onde  $(\xi, \eta)$  é a solução exata do sistema e  $(b_0, c_0)$  são valores iniciais que podem ser obtidos por um método mapeamento descrito em I.4.

Usando o desenvolvimento em série de Taylor no sistema (A4.08) tem-se:

$$\left\{ \begin{array}{l} f(b_n + \Delta b, c_n + \Delta c) = f(b_n, c_n) + \Delta b \cdot f_b(b_n, c_n) + \Delta c \cdot f_c(b_n, c_n) + \dots = 0 \\ \varphi(b_n + \Delta b, c_n + \Delta c) = \varphi(b_n, c_n) + \Delta b \cdot \varphi_b(b_n, c_n) + \Delta c \cdot \varphi_c(b_n, c_n) + \dots = 0 \end{array} \right. \quad (A4.14)$$

nesta expressão  $f_b$ ,  $f_c$ ,  $\varphi_b$  e  $\varphi_c$  correspondem às derivadas parciais:

$$f_b = \frac{\partial f(b, c)}{\partial b}; \quad f_c = \frac{\partial f(b, c)}{\partial c}; \quad \varphi_b = \frac{\partial \varphi(b, c)}{\partial b} \quad \text{e} \quad \varphi_c = \frac{\partial \varphi(b, c)}{\partial c}$$

Considerando somente os termos de primeira ordem de (A4.14) pode-se aproximar o sistema por um sistema linear:

$$\left\{ \begin{array}{l} \Delta b \cdot f_b + \Delta c \cdot f_c \cong -f \\ \Delta b \cdot \varphi_b + \Delta c \cdot \varphi_c \cong -\varphi \end{array} \right. \quad (A4.15)$$

e resolvendo-se este sistema linear, em  $\Delta b$  e  $\Delta c$ , tem-se:

$$\begin{cases} \Delta b = \frac{\varphi \cdot f_c - f \cdot \varphi_c}{f_b \cdot \varphi_c - \varphi_b \cdot f_c} \\ \Delta c = \frac{f \cdot \varphi_b - \varphi \cdot f_b}{f_b \cdot \varphi_c - \varphi_b \cdot f_c} \end{cases} \quad (A4.16)$$

substituindo agora esta solução em (A4.13) tem-se o seguinte algoritmo:

$$\begin{cases} b_{n+1} = b_n + \frac{\varphi(b_n, c_n) \cdot f_c(b_n, c_n) - f(b_n, c_n) \cdot \varphi_c(b_n, c_n)}{J(b_n, c_n)} \\ c_{n+1} = c_n + \frac{f(b_n, c_n) \cdot \varphi_b(b_n, c_n) - \varphi(b_n, c_n) \cdot f_b(b_n, c_n)}{J(b_n, c_n)} \end{cases} \quad (A4.17)$$

onde o Jacobiano é dado por:

$$J(b_n, c_n) = \begin{vmatrix} \frac{\partial f(b_n, c_n)}{\partial b} & \frac{\partial f(b_n, c_n)}{\partial c} \\ \frac{\partial \varphi(b_n, c_n)}{\partial b} & \frac{\partial \varphi(b_n, c_n)}{\partial c} \end{vmatrix} \quad (A4.18)$$

#### I.4 - Busca da solução aproximada - par ordenado (bo, co)

Como as equações do sistema mudam de acordo com a região em que **b** e **c** se encontram, deve-se resolver quatro sistemas de equações e o programa segue os seguintes passos:

##### Passo 1

Admite-se um incremento  $\delta$ , onde  $0 < \delta < 1$ , e o valor deste varia de acordo com a região a ser pesquisada. Faz-se um mapeamento das funções **F(b,c)=0** e **φ(b,c)=0** em cada região separadamente utilizando o incremento  $\delta$  como passo para **b** e **c**. Após o mapeamento encontram-se quatro vetores dados por (A4.19)

$$F(Bk_f, Ck_f) \Rightarrow \begin{cases} B_f = [B1_f, B2_f, B3_f, \dots, Bm_f] \\ C_f = [C1_f, C2_f, C3_f, \dots, Cm_f] \end{cases} \quad (A4.19)$$

$$\varphi(Bk_\varphi, Ck_\varphi) \Rightarrow \begin{cases} B_\varphi = [B1_\varphi, B2_\varphi, B3_\varphi, \dots, Bn_\varphi] \\ C_\varphi = [C1_\varphi, C2_\varphi, C3_\varphi, \dots, Cn_\varphi] \end{cases}$$

## Passo 2

Estabelece-se um limiar  $\Delta > 0$ , próximo de zero, e obtém-se vetores **b** e **c** reduzidos de (A4.19) cujos valores absolutos das funções **F(B,C)** e **φ(B,C)** são menores que o limiar adotado.

$$|F(bk_f, ck_f)| \leq \Delta \Rightarrow \begin{cases} b_f = [b1_f, b2_f, b3_f, \dots, bp_f] \\ c_f = [c1_f, c2_f, c3_f, \dots, cp_f] \end{cases} \quad (A4.20)$$

$$|\varphi(Bk_\varphi, Ck_\varphi)| \leq \Delta \Rightarrow \begin{cases} b_\varphi = [b1_\varphi, b2_\varphi, b3_\varphi, \dots, bq_\varphi] \\ c_\varphi = [c1_\varphi, c2_\varphi, c3_\varphi, \dots, cq_\varphi] \end{cases}$$

com  $p < m$  e  $q < n$

## Passo 3

Com os vetores reduzidos **b** e **c**, neste passo deve-se verificar quais são as coordenadas comuns que simultaneamente tornam  $F(b_i, c_i) \cong 0$  e  $\varphi(b_i, c_i) \cong 0$ . Isto é realizado comparando-se as componentes dos vetores (**bf**, **bφ**) e (**cf**, **cφ**). Como resultado obtém-se um par de vetores reduzidos de (A4.20) representados em (A4.21):

$$\begin{aligned} b &= [b_0 \ b_1 \ b_2 \ b_3 \dots b_k] \\ c &= [c_0 \ c_1 \ c_2 \ c_3 \dots c_k] \end{aligned} \quad (A4.21)$$

Observa-se que para que estes vetores sejam reduzidos deve-se ter:  $k < p$  e  $k < q$ . Os vetores **b** e **c** têm como elementos os valores iniciais (**bo**, **co**) desejados.

## I.5 Resolução final do sistema



Após o mapeamento de cada região, pode-se obter os vetores **b** e **c** com dimensão maior que um, o que poderá resultar nas seguintes soluções finais:

**a - Todos pares ordenados  $(b_i, c_i)$  convergem para o mesmo valor de raiz**

Neste caso toma-se esta como solução única.

**b - Alguns pares ordenados convergem para uma solução e outros convergem para outra solução**

Este caso é ilustrado na fig. A4.03.

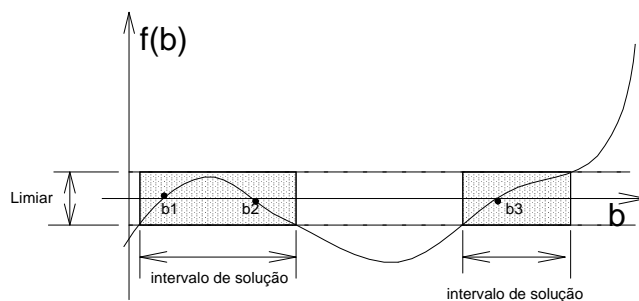


fig. A4.03

Afim de resolver os itens **a** e **b** descritos lança-se mão do seguinte processo:

Dados os vetores soluções iniciais no primeiro intervalo:

$$\begin{aligned} b &= [b_0 \ b_1 \ b_2 \ b_3 \ \dots b_k] \\ c &= [c_0 \ c_1 \ c_2 \ c_3 \ \dots c_k] \end{aligned} \tag{A4.22}$$

aplica-se o método de NR para cada par ordenado  $(b_i, c_i)$  e encontram-se as soluções:

$$\begin{aligned}
(b_i, c_i) &\rightarrow (\xi_i, \eta_i) \\
(b_{i+1}, c_{i+1}) &\rightarrow (\xi_i, \eta_i) \\
&\vdots \\
(b_k, c_k) &\rightarrow (\xi_k, \eta_k) \neq (\xi_i, \eta_i)
\end{aligned} \tag{A4.23}$$

Observa-se que podem existir dois conjuntos de raízes distintas  $[(\xi_i, \eta_i), (\xi_k, \eta_k)]$ . Para verificar esta possibilidade adota-se um novo limiar  $\Delta x$  e compara-se com o valor da diferença  $\Delta x = \|\xi_i\| - \|\xi_k\|$ . Assim, se:

$$\begin{cases} \Delta x \leq \text{limiar} \rightarrow \begin{cases} \xi_i = \xi_k \text{ e } \eta_i \neq \eta_k \\ \xi_i = \xi_k \text{ e } \eta_i = \eta_k \end{cases} \rightarrow \text{única solução} \\ \Delta x > \text{limiar} \rightarrow \begin{cases} \xi_i \neq \xi_k \text{ e } \eta_i \neq \eta_k \\ \xi_i \neq \xi_k \text{ e } \eta_i = \eta_k \end{cases} \end{cases} \tag{A4.24}$$

Logo pode-se obter, para cada região, uma ou mais raízes, dependendo do resultado obtido em (A4.24).