Apêndice A

MMPP de 2 estados

Neste apêndice são obtidas expressões exatas do número esperado de chegadas no intervalo $(0, t)$, sua variância e seu terceiro momento para um processo de Poisson modulado por Markov (MMPP) de 2 estados [9]. O MMPP de 2 estados é um processo duplamente estocástico, onde a taxa do processo de chegada varia de acordo com o estado de uma cadeia de Markov de 2 estados. Sejam $1/r_i$ o tempo médio de permanência provisória no estado i $(i = 1, 2)$ e λ_i a respectiva taxa de chegada do processo de Poisson para este estado. Seja π o vetor das probabilidades de equilíbrio da cadeia de Markov. Assim, a função de geração de probabilidade do número de chegadas em um intervalo de comprimento t é

$$g(z,t) = \pi \exp\left\{\left[Q + (z+1)\Lambda\right] t\right\} e \ , \tag{A.1}$$

onde, para o MMPP de 2 estados,

$$\pi = \tfrac{1}{r_1+r_2}\begin{pmatrix} r_1 & r_2 \end{pmatrix} \ , \qquad e = \begin{pmatrix} 1 & 1 \end{pmatrix}^T \ ,$$

$$Q = \begin{bmatrix} -r_1 & r_1 \\ r_2 & -r_2 \end{bmatrix} \qquad \text{e} \qquad \Lambda = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \ ,$$

Definindo N_t como o número de chegadas em $(0, t)$ e $\overline{N_t} = E[N_t]$, tem-se por [17] que

$$\overline{N_t} = \pi\lambda t = \frac{\lambda_1 r_2 + \lambda_2 r_1}{r_1 + r_2} t \ ; \tag{A.2}$$

onde

$$\lambda = \begin{bmatrix} \lambda_1 & \lambda_2 \end{bmatrix} \ .$$

A razão variância-sobre-média de N_t, dada em [9], é

$$\frac{\text{var}[N_t]}{N_t} = 1 + \frac{2(\lambda_1 - \lambda_2)^2 r_1 r_2}{(r_1 + r_2)^2(\lambda_1 r_2 + \lambda_2 r_1)} - \frac{2(\lambda_1 - \lambda_2)^2 r_1 r_2}{(r_1 + r_2)^3(\lambda_1 r_2 + \lambda_2 r_1)t}\left(1 - e^{-(r_1+r_2)t}\right) \quad (A.3)$$

O terceiro momento de N_t é dado por

$$E\left[\left(N_t - \overline{N_t}\right)^3\right] = \left.\frac{\delta^3 g(z,t)}{\delta z^3}\right|_{z=1} - 3\overline{N_t}(\overline{N_t} - 1)\frac{\text{var}(N_t)}{\overline{N_t}} - \overline{N_t}(\overline{N_t} - 1)(\overline{N_t} - 2) \quad (A.4)$$

Para obter a terceira derivada requerida em (A.4), deve-se lembrar que

$$g^{(3)}(1,t) = \left.\frac{\delta^3 g(z,t)}{\delta z^3}\right|_{z=1} = L^{-1}\left[\left.\frac{\delta^3 \tilde{g}(z,s)}{\delta z^3}\right|_{z=1}\right] \quad (A.5)$$

onde

$$\tilde{g}(z,s) = \pi\left[sI - Q - (z-1)\Lambda\right]^{-1} e \quad (A.6)$$

é a transformada de Laplace de $g(z,t)$ e L^{-1} denota a transformada inversa de Laplace. Assim, tem-se

$$\begin{aligned} g^{(3)}(1,t) &= L^{-1}\left\{6\pi\left[(sI-Q)^{-1}\Lambda\right]^3 (sI-Q)^{-1}e\right\} \\ &= L^{-1}\left\{\frac{6}{s^2}\pi\Lambda\left[(sI-Q)^{-1}\Lambda\right]^2 e\right\} \end{aligned} \quad (A.7)$$

que após a inversão resulta em

$$g^{(3)}(1,t) = \frac{6}{r_1 + r_2}\left[\frac{A_{11}}{6}t^3 + \frac{A_{21}}{2}t^2 + A_{31}t + A_{12}te^{-(r_1+r_2)t} + A_{41}\left(1 - e^{-(r_1+r_2)t}\right)\right] \quad (A.8)$$

onde

$$A_{11} = \frac{(\lambda_1 r_2 + \lambda_2 r_1)^3}{(r_1 + r_2)^2},$$

$$A_{21} = \frac{2r_1 r_2(\lambda_1 - \lambda_2)^2(\lambda_1 r_2 + \lambda_2 r_1)}{(r_1 + r_2)^3},$$

$$A_{31} = \frac{r_1 r_2(\lambda_1 - \lambda_2)^2[\lambda_1 r_1 + \lambda_2 r_2 - 2(\lambda_1 r_2 + \lambda_2 r_1)]}{(r_1 + r_2)^4},$$

$$A_{41} = \frac{-2r_1 r_2(\lambda_1 - \lambda_2)^3(r_1 - r_2)}{(r_1 + r_2)^5},$$

e finalmente

$$A_{21} = \frac{r_1 r_2(\lambda_1 - \lambda_2)^2(\lambda_1 r_1 + \lambda_2 r_2)}{(r_1 + r_2)^4}.$$

Apêndice B

Programa de Simulação

Neste apêndice, é apresentada a listagem do programa utilizado para a simulação da mistura de fontes em rajadas do tipo silêncio/atividade com fontes de vídeo. A linguagem utilizada é própria para o simulador QNAP2.

```
/DECLARE/

& variaveis para o calculo do intervalo de confianca
INTEGER i, k;
REAL num_viol, p_viol, dif, h, tvar, viol(10), n_gera;

& parametros do multiplexador
QUEUE BIFURCA, MULT;
INTEGER Max_Mult, Num_Fila, ncel, nbytes;
REAL f_enlace;

& parametros da fonte de video
QUEUE RAJADAS;
INTEGER raj_comp, N_Video;
REAL atraso;
REAL tx_atual, alpha, beta, ni;
REAL T_quadro, tx_N;
CUSTOMER REAL tx_ant, T_celula, sobra;
```

```
& parametros da fonte em rajada
QUEUE INATIV, ATIVA;
INTEGER n_ativas, n_inativ, N_Voz;
INTEGER L;
REAL taxa_off, taxa_on;
REAL duracao, D, S, Bp, b, delta, f_rajad;

& Dados do usuario comum a todos os tipos de fontes
CUSTOMER INTEGER cont, comp;
REF CUSTOMER celula;

& Fonte de Video
/STATION/ NAME = RAJADAS;
          TYPE = MULTIPLE(N_Video);
          INIT = N_Video;
          SERVICE = BEGIN
          IF (TIME = 0) THEN BEGIN
             CUSTOMER.tx_ant := 0.52;
             CUSTOMER.sobra := 0;
             atraso := UNIFORM (0 , T_quadro);
             CST (atraso);
          END;
          tx_atual := alpha * CUSTOMER.tx_ant + beta * NORMAL (ni,1);
          IF (tx_atual < 0) THEN
          BEGIN
             tx_atual := 0;
             CUSTOMER.tx_ant := tx_atual;
             CST(T_quadro);
          END
          ELSE BEGIN
             tx_N := INTREAL (tx_atual * 2.5 * 10**5) + CUSTOMER.sobra;
             raj_comp := INTREAL ( tx_N / (48 * 8) );
             CUSTOMER.tx_ant := tx_atual;
             IF (raj_comp = 0) THEN
             BEGIN
                CUSTOMER.sobra := tx_N;
```

```
                CST(T_quadro);
             END
             ELSE
             BEGIN
                CUSTOMER.sobra := tx_N - (raj_comp * 48 * 8);
                CUSTOMER.T_celula := T_quadro / raj_comp;
                CUSTOMER.comp := raj_comp;
                CUSTOMER.cont := 0;
                WHILE (CUSTOMER.cont < CUSTOMER.comp) DO
                BEGIN
                   celula := NEW(CUSTOMER);
                   TRANSIT(celula,BIFURCA);
                   CUSTOMER.cont := CUSTOMER.cont + 1;
                   CST(T_celula);
                END;
             END;
          END;
          TRANSIT(RAJADAS);
END;

& Fonte de Voz
STATION/ NAME = ATIVA;
          TYPE = MULTIPLE(N_Voz);
          INIT = n_ativas;
          SERVICE = BEGIN
             duracao := EXP(1/taxa_on);
             CUSTOMER.comp := INTREAL(duracao * f_rajad);
             CUSTOMER.cont := 1;
             WHILE (CUSTOMER.cont <= CUSTOMER.comp) DO
             BEGIN
                n_gera := n_gera + 1;
                celula := NEW(CUSTOMER);
                TRANSIT(celula,BIFURCA);
                CST(delta);
                CUSTOMER.cont := CUSTOMER.cont + 1;
             END;
```

```
          END;
          TRANSIT = INATIV;

/STATION/ NAME = INATIV;
          TYPE = MULTIPLE(N_Voz);
          INIT = n_inativ;
          SERVICE = BEGIN
             EXP(1/taxa_off);
          END;
          TRANSIT = ATIVA;

& Verificador de descartes
/STATION/ NAME = BIFURCA;
          SERVICE = BEGIN
          n_gera := n_gera + 1;
          IF (Num_Fila < Max_Mult) THEN
             Num_Fila := Num_Fila + 1
          ELSE
             num_viol := num_viol + 1;
          END;
          TRANSIT = OUT;

& Multiplexador
/STATION/ NAME = MULT;
          INIT = 1;
          SERVICE = BEGIN
          CST(1/f_enlace);
          IF Num_Fila > 0 THEN
             Num_Fila := Num_Fila - 1;
          END;
          TRANSIT = MULT;

/CONTROL/
          TMAX = 300;
          ESTIM = REPLICATION(1);
          ACCURACY = ALL QUEUE;
```

```
          &OPTION = TRACE;


/EXEC/ BEGIN

& INICIALIZA PARAMETROS DO MODELO

& Parametros do Multiplexador
          nbytes := 53;
          ncel := nbytes * 8;
          Max_Mult := 35;
          f_enlace := (150*10**6)/ncel;

& Parametros da Fonte de Video
          N_Video := 1;
          alpha := 0.8781;
          beta := 0.1108;
          ni := 0.572;
          T_quadro := 1./30;

& Parametros da Fonte de Video
          N_Voz := 60;
          b := 10.0;
          Bp := 10*10**6;
          L := 100;
          delta := ncel/Bp;
          D := L*delta;
          S := D*(b-1);
          taxa_off := 1.0/S;
          taxa_on := 1.0/D;
          f_rajad := Bp/ncel;

          PRINT();
          PRINT("Numero de Fontes de Video Simuladas:  ", N_Video);
          PRINT("Numero de Fontes de Voz Simuladas:  ", N_Voz);
          PRINT(); PRINT();
```

```
& CALCULA PROBABILIDADE DE VIOLACAO COM 10 REPLICACOES:
          p_viol := 0;
          FOR k := 1 STEP 1 UNTIL 10 DO
          BEGIN
             & reinicializa os parametros do multiplexador
             n_gera := 0;
             Num_Fila := 0;
             num_viol := 0.0;
             & seta o numero inicial de fontes de voz
             n_ativas := RINT(0,N_Voz);
             n_inativ := N_Voz - n_ativas;
             PRINT("Numero de Fontes Ativas:  ", n_ativas);
             PRINT("Numero de Fontes Inativas:  ", n_inativ);
             & inicializa a simulacao
             SIMUL;
             viol(k) := num_viol/n_gera;
             p_viol := p_viol + viol(k);
             PRINT("Numero de Celulas Violadoras :",num_viol);
             PRINT("Numero de Celulas Geradas :", n_gera);
             PRINT ("Probabilidade de Violacao Parcial = ", viol(k), "k =", k);
          END;

& CALCULA INTERVALO DE CONFIANCA

          p_viol := p_viol/10.0;
          tvar := 0.0;
          FOR i := 1 STEP 1 UNTIL 10 DO
          BEGIN
             dif := (viol(i) - p_viol);
             PRINT ("i = ", i, "viol(i) = ", viol(i));
             tvar := tvar + (dif**2);
          END;
          tvar := tvar/9.0;
          PRINT("Variancia = ", tvar);
          h := 2.23 * SQRT(tvar/10.0);
```

```
          PRINT("Probabilidade de Violacao = ", p_viol);
          PRINT("Intervalo de Confianca = ", h);
END;

/END/
```