

ANPRO: ANALISADOR AUTOMÁTICO DE PROTOCOLOS DE COMUNICAÇÃO
MODELADOS EM RPN (REDE DE PETRI NUMÉRICA)

Berenice C. Damasceno, Walter C. Borelli

Depto. de Telemática
FEE/UNICAMP
13081, Campinas, SP

SUMÁRIO

Nesse trabalho utilizamos um método formal de validação de protocolos de comunicação baseado no modelo de transição. Tal método utiliza o modelo da Rede de Petri Numérica (RPN) ("NUMERICAL PETRI NET") [4,9]. Escolhemos a RPN por ser um modelo mais geral e de maior poder descritivo que a rede de Petri Clássica [8]. Com esse modelo definido, desenvolvemos uma ferramenta de análise automática de protocolos de comunicação modelados em RPN, o ANPRO [5]. Dessa forma, propomos uma metodologia de análise para validação de protocolos utilizando tal ferramenta e apresentamos um exemplo.

1. INTRODUÇÃO

A validação de protocolos pode ser entendida de várias formas: a verificação de propriedades do protocolo, a verificação de que o protocolo satisfaz a especificação do serviço e a verificação de que um determinado programa de fato implementa o protocolo corretamente [5].

A maior parte dos trabalhos que têm sido feitos [2] se concentra na verificação de propriedades do protocolo. Dentre as propriedades podemos citar ausência de impasses ("deadlock"), ausência de loop, exclusão mútua de seções críticas, etc.

Os métodos baseados em modelos de transição fazem a validação basicamente através da geração sistemática de todos os estados possíveis do sistema. A maior dificuldade encontrada por estes métodos é a grande quantidade de estados que pode ocorrer na validação de determinados protocolos.

Os métodos baseados em linguagens de programação empregam técnicas usadas em verificação de programas. Estas técnicas podem, potencialmente, lidar com todos os tipos de propriedades que se deseje provar, mas a sua automação é bastante difícil.

Os métodos híbridos usam uma combinação das duas técnicas descritas acima; o uso de variáveis simbólicas, por exemplo, permite a redução do número total de estados possíveis do sistema. Estas técnicas são difíceis de automatizar completamente.

Neste trabalho o método formal de validação de protocolo de comunicação é baseado no modelo de transição. O modelo utilizado foi a Rede de Petri Numérica (RPN), que é uma extensão da Rede de Petri, sendo este um formalismo que tem sido muito usado para modelamento e análise de protocolos [6,7]. Isto se deve ao fato que desde sua criação a Rede de Petri tem sido muito usada na descrição e análise de sistemas concorrentes assíncronos que se comunicam, devido à sua simplicidade de funcionamento e à sua característica gráfica que permite uma visualização fácil dos sistemas modelados.

Apresentamos aqui uma ferramenta de análise automática de protocolos de comunicação modelados em RPN, o ANPRO (Analisador de PROtocolos de Comunicação). Apresentamos o algoritmo para sua implementação e analisamos um exemplo de protocolo modelado em RPN utilizando este pacote de "software".

2. DESCRIÇÃO DA RPN

2.1. DEFINIÇÕES

Uma RPN pode ser definida pela óctupla:

$RPN = (L, T, H, CD, RD, OP, P, EI)$, onde:

- L : é o conjunto de lugares (círculos) com um número inteiro positivo de elementos
- T : é o conjunto de transições (barras) com um número inteiro positivo de elementos;
- H : representa a condição de habilitação dos arcos de entrada das transições;
- CD : representa as condições de disparo dos arcos de entrada das transições;
- RD : representa as regras de pós-disparo das transições, definindo quais senhas serão colocadas nos respectivos lugares de saída;
- OP : representa as operações ou ações realizadas pela transição, fornecendo uma visão do que ocorre no protocolo quando determinada transição dispara na RPN;
- P : representa os predicados, ou seja, as condições de habilitação, além das anteriores que sejam também relevantes. Como por exemplo pode ser uma relação entre senhas ou variável que distinga duas transições aparentemente iguais (mesmos lugares de entrada e condições). Estas condições são colocadas entre colchetes ao lado de cada transição.
- EI : é o estado inicial da RPN. É constituído por:
 - a) uma marcação inicial M_0 que é um conjunto finito e não vazio das condições iniciais, onde são discriminados os lugares que possuam senhas ("token") e de que tipos são tais senhas S.
 - b) o conjunto dos valores iniciais das variáveis relacionadas aos predicados e operações das transições, ou seja:
 $EI = (M_0, VI)$, onde

Mø = marcação inicial

VI = valores iniciais das variáveis relativas aos predicados e operações da transição e,

S = representa os diferentes tipos de senhas da RPN, ou seja, cada tipo de senha representa um número inteiro positivo ou um caracter ou uma palavra.

A Fig. 1¹ apresenta as notações e a forma gráfica a ser usada neste trabalho para apresentação da RPN.

Para todo arco de entrada de uma transição existe uma condição de habilitação H referente às senhas que estão no lugar de entrada, ao qual o arco de entrada é conectado. Nesta Fig. 1 temos as condições H1, H2, H5 e H7. Como exemplo podemos ter que para a transição T2 ficar habilitada, a condição H1 deve ser igual a S1, ou seja, é necessário que no lugar L1 exista pelo menos uma senha do tipo S1, onde S1 pode significar sinal de reconhecimento, por exemplo.

Para todo arco de entrada existe uma condição de disparo CD, a qual define quais senhas serão removidas do respectivo lugar de entrada quando a transição disparar. Neste exemplo temos as condições de disparo de entrada CD1 e CD2 da transição T2, CD5 da transição T1 e CD7 da transição T3.

Para todo arco de saída de uma transição existe uma regra de pós-disparo RD, que define quais senhas serão colocadas no respectivo lugar de saída após o disparo da transição. Nesta Fig. 1 temos RD3 e RD4 da transição T2, RD6 da transição T1 e RD8 da transição T3 como regras de pós-disparo.

Para toda transição pode existir uma lista de operações ou ações, OP, a qual é colocada no lado da transição como pode ser visto neste exemplo, na transição T2, onde OP é " $Y = F(x)$ ".

Além disso para toda transição pode existir condições de habilitação (além das condições H), P, que também possam ser relevantes. Tais condições P podem representar relações entre senhas ou variáveis, por exemplo, e devem ser colocadas entre colchetes ao lado da transição correspondente, como na Fig. 1, onde temos $[Z=K]$, ao lado de T1.

Quando todas as condições de habilitação (H e P) forem verificadas e as de disparo (CD) também, referentes à uma determinada transição, esta pode disparar.

Como ocorre nas RP Clássicas, nas RPN's somente uma transição dispara por vez, e quando isso ocorre segue-se um conjunto de ações que é executado na seguinte seqüência:

- 1 - senhas(s) são removidas dos lugares de entrada de acordo com as condições de disparo (CD);
- 2 - o resultado da(s) operação(ões) na transição é colocado na memória de dados da rede, atualizando-a e,
- 3 - as senhas são colocadas nos lugares de saída de acordo com as regras de pós-disparo da transição (RD).

Observe que as senhas podem ter qualquer natureza e valor, onde as mais simples seriam os números naturais, como nas RP Clássicas.

A cada tipo de senha é relacionado um temporizador que realiza a função de controle desta. Seu principal objetivo é o de evitar e/ou localizar situações de perda de sinais, controlar o número de retransmissões de um sinal e influenciar no disparo de transições.

¹ As figuras e tabelas encontram-se ao final do artigo.

2.2. PROPRIEDADES DA RPN

As principais propriedades da RPN são apresentadas a seguir, baseando-se nas propriedades da RP, pois aproveitamos uma proporção considerável da teoria da RP [8] para o modelo da RPN.

2.2.1. Limitabilidade

Uma RPN é definida como limitada a um valor n (n inteiro) se para qualquer marcação M decorrente pertencente ao estado decorrente E da marcação inicial M_0 pertencente ao estado E_0 , o número de cada tipo de senha S em cada lugar l_j pertencente a $L, j = 1, 2, \dots, NL$ for sempre menor que n .

Numa RPN limitada, o número de estados é finito. Por outro lado, uma RPN não limitada possui infinitos estados, significando que o sistema físico correspondente é impossível de ser implementado, analogamente ao que ocorre quando da utilização da Rede de Petri Clássica.

2.2.2. Vivacidade

Uma RPN é dita viva para um estado inicial E_0 , se para qualquer estado decorrente E existir uma seqüência de disparos τ que torne disparável qualquer transição da rede.

Matematicamente,

$$\forall E \in \{D(E_0)\} ; \forall t \in T$$

$$\exists \tau / (E \xrightarrow{\tau'} E' \xrightarrow{t} E'')$$

onde $\tau = \{t', t\}$, $E', E'' \in \{D(E_0)\}$.

τ' é uma seqüência de disparos que leva o estado E ao estado E' ;

T é o conjunto de todas as transições da rede;

$D(E_0)$ é o conjunto de todos os estados pelos quais o sistema pode passar a partir do estado inicial E_0 .

Um sistema físico representado por uma RPN viva é livre de impasses.

2.2.3. Reiniciabilidade

Uma RPN é reiniciável para um dado estado inicial E_0 , se para qualquer estado decorrente E pertencente a $\{D(E_0)\}$ existe uma seqüência de disparos \emptyset que faça a rede voltar ao estado inicial.

Matematicamente,

$$\forall E \in \{D(E_0)\} , \exists \emptyset / (E \xrightarrow{\emptyset} E_0)$$

Um sistema físico, representado por uma RPN reiniciável tem características de retornar ao seu estado inicial após a execução de uma ou mais tarefas. Esta, geralmente, se constitui uma condição necessária para o bom funcionamento de vários sistemas, analogamente à RP.

3. ANALISADOR AUTOMÁTICO DE PROTOCOLOS DE COMUNICAÇÃO: ANPRO

O ANPRO foi desenvolvido em linguagem PASCAL (TURBO - versão 3.0), é executado em microcomputador tipo PC/aT e utiliza alocação dinâmica de memória.

O ANPRO é constituído por três sub-programas, o DESCRPN, que trata da descrição e documentação da RPN; o ANASIM, que faz a análise da rede e/ou uma simulação (iterativa com o usuário) e o DOCRPN, que documenta os resultados obtidos pelo ANASIM. A Fig. 2 apresenta o diagrama modular do ANPRO.

A seguir apresentamos cada sub-programa.

3.1. SUBPROGRAMA DESCRPN

O sub-programa DESCRPN através de um modo interativo com o usuário fornece as opções de descrição, alteração e documentação da RPN.

O diagrama modular da Fig. 3 mostra os menus que compõem o subprograma DESCRPN, detalhando cada opção deste.

Podem ocorrer casos onde há apenas a utilização de um só tipo de senha, principalmente em protocolos menos complexos, para tais casos é possível ao usuário optar, se desejar, por apenas um tipo de senha (valor default). Caso contrário pode optar por no máximo cinco tipos de senhas, um número razoável diante dos protocolos que já verificamos, mas que pode ser aumentado sem nenhuma dificuldade. Em seguida, o usuário deverá digitar o número da transição e seus dados de entrada, que são as condições de disparo (CD) e os predicados (P). A seguir, deverá digitar os dados de saída da transição, que são as regras de pós-disparo (RD) e as operações (OP). Após as definições de todas as transições, o usuário poderá ainda optar pela atribuição de nomes aos lugares, às transições, às senhas e para fins de documentação ele poderá relacionar as operações que são realizadas após o disparo das transições com relação ao protocolo.

PREDICADOS

No caso dos predicados associados às transições da RPN para fins de análise, consideramos os seguintes tipos (a definição de outros tipos é análoga e perfeitamente viável):

- (1) $X_i + Y_i \leq K_i$, soma de duas variáveis inteiras, X_i e Y_i , comparadas com o valor de variável inteira K_i , onde $i \in [1,5]$. As variáveis X_i , Y_i , K_i têm seus valores guardados na memória de dados (MD) da RPN. O caso mais comum é:

$X_i + 1 \leq K_i$, incremento de um número inteiro de uma unidade e de sua comparação com o valor de K_i , o que pode implicar uma limitação do número de disparos de uma transição.

- (2) $X_i - Y_i \geq K_i$, diferença de duas variáveis inteiras, X_i e Y_i , comparadas com o valor de variável inteira, K_i . O caso mais comum é:
 $X_i - 1 \geq K_i$, ou seja, decrementar um número inteiro de uma unidade e comparar com K_i .

- (3) $X_i=0$, predicado que verifica se o valor da variável X_i é 'zero'.
- (4) $X_i=1$, predicado que verifica se o valor da variável X_i é 'um'.
- (5) $X_i > 0$, predicado que verifica se o valor da variável X_i é maior que 'zero'.
- (6) $X_i > K_i$, predicado que verifica se o valor da variável X_i é maior que o da variável K_i .
- (7) $T_i=0$, predicado que verifica se o temporizador T_i , onde $i \in [1,5]$, está desativado (OFF).
Obs. Os temporizadores associados à RPN por default iniciam com valor zero (desativados) e, por opção do usuário poderão não ser utilizados.
- (8) $T_i=1$, predicado que verifica se o temporizador T_i está ativado (ON).

OPERAÇÕES

No caso das operações (OP) das transições da RPN colocamos seis opções ao usuários (a definição de outras operações é análoga e perfeitamente viável):

- (1) X_i+Y_i , soma de duas variáveis inteiras, X_i e Y_i . O caso mais comum é: X_i+1 , ou seja, incrementar um número inteiro de uma unidade.
- (2) X_i-Y_i , diferença de duas variáveis inteiras, X_i e Y_i . O caso mais comum é: X_i-1 , ou seja, decrementar um número inteiro de uma unidade.
- (3) $X_i=0$, atribuição do valor 'zero' à variável X_i .
- (4) $X_i=1$, atribuição do valor 'um' à variável X_i .
- (5) $T_i=1$, significa a inicialização do temporizador T_i .
- (6) $T_i=0$, significa a interrupção ou vencimento do temporizador T_i .

ALTERAÇÕES

Após a descrição da RPN, pode-se realizar várias alterações, bastando para isso escolher a opção "Alteração" (Fig. 3).

DOCUMENTAÇÃO

Finalmente, após a RPN ser descrita (e/ou alterada) deve-se armazená-la num arquivo através da opção "Documentação" (Fig. 3) para o seu interfaceamento com o ANASIM.

3.2. SUBPROGRAMA ANASIM

O subprograma ANASIM pode ser definido, em termos de análise de protocolos, como sendo o principal constituinte do ANPRO. Ele realiza a análise da rede e/ou simulação do protocolo (ou parte do protocolo), considerando suas propriedades e fornecendo dados úteis a possíveis verificações ou correções do protocolo analisado, que são documentados posteriormente pelo DOCRPN.

Este subprograma permite montar a tabela de estados, que é composta pela tabela de marcações e pela memória de dados da rede; o grafo de estados e a tabela das componentes fortemente conexas; permite verificar as propriedades da RPN e permite ao usuário realizar uma simulação do protocolo modelado em RPN escolhendo a seqüência de disparo das transições que desejar.

Ao iniciar a análise, o usuário deve fornecer o nome do arquivo onde está a descrição da rede criada pelo subprograma DESCRPN.

Em seguida deve digitar o número máximo de estados para análise.

Ao verificar a ocorrência do primeiro estado superior (rede não limitada) a execução do programa é interrompida e o usuário deve optar por sua continuação ou interrupção. Em ambos os casos o usuário é avisado que não serão verificadas as propriedades de vivacidade e reiniciabilidade, pois a tabela e o grafo de estados estarão incompletos.

Em seguida é pedido o nome do arquivo para armazenar o resultado da análise que poderá ser documentada pelo DOCRPN.

Se o usuário desejar realizar uma simulação do protocolo deverá fornecer o nome do arquivo onde se encontra a descrição da RPN criado pelo DESCRPN.

A seguir é mostrado ao usuário no terminal de vídeo as transições que podem disparar no estado inicial e, então ele deve digitar o número da transição escolhida para disparar, gerando um novo estado e outras opções de transições que podem disparar.

Esse procedimento deve ser repetido até que o usuário decida pela sua interrupção ou até que o número máximo de estados seja atingido. É conveniente salientar que os dados para a simulação podem ser fornecidos via arquivo apropriado.

Em seguida é pedido o nome do arquivo para armazenar a simulação que poderá ser documentada pelo DOCRPN.

A simulação fornece ao usuário a tabela e o grafo de estados correspondentes às suas opções de disparo das transições da rede.

A Fig. 4 mostra o diagrama modular do subprograma ANASIM.

3.3. SUBPROGRAMA DOCRPN

Este subprograma permite realizar a documentação dos resultados da análise e/ou simulação da RPN. Ao iniciar a documentação, o usuário deve fornecer o nome do arquivo onde está o resultado da análise e/ou simulação da RPN.

Este subprograma foi estruturado conforme a Fig. 5. Esta figura é bem detalhada, fazendo com que seja bem simples verificar as opções de que o usuário dispõe.

Sendo assim, passamos agora para a utilização deste pacote na análise de validação de um exemplo de protocolo de comunicação.

4. METODOLOGIA DE ANÁLISE PARA VALIDAÇÃO DE PROTOCOLOS DE COMUNICAÇÃO UTILIZANDO O ANPRO

A metodologia de análise para a validação de protocolos usando o ANPRO consiste das seguintes etapas:

- a) Converte-se o protocolo de uma especificação em linguagem natural para o modelo da RPN. Se o protocolo estiver especificado em uma linguagem formal de projeto (diagrama de estados, SDL: Functional Specification and Description Language - CCITT [3], etc.) converte-se o protocolo para o

modelo da RPN equivalente. Tal conversão deve garantir a manutenção da integridade do protocolo.

- b) Através do ANPRO, verificamos as propriedades específicas da RPN (limitabilidade, reiniciabilidade, vivacidade).
- c) A não verificação (imperfeição) de uma de suas propriedades pode ser indicativo da existência de erros na especificação do protocolo de um modelamento em RPN não satisfatório.

A análise de um mau dimensionamento da RPN equivalente e/ou de falhas de projeto do protocolo podem em muito ser enriquecidas por diagnósticos mais detalhados, que o ANPRO pode fornecer sobre as imperfeições da rede. Feitas as modificações necessárias, volta-se à etapa anterior e assim até a correção total das eventuais falhas do protocolo.

4.1. UTILIZAÇÃO E INTERPRETAÇÃO DOS DIAGNÓSTICOS FORNECIDOS PELO ANPRO

Através dos diagnósticos fornecidos pelo subprograma DOCRPN (Fig. 5), passamos a interpretá-los e analisá-los. Sendo assim colocamos a seguir os diagnósticos possíveis e suas interpretações:

- A) A rede é não limitada: isto significa que existe no mínimo um estado superior a um outro em um de seus ramos ascendentes. Assim existe uma seqüência de ações (seqüência de disparo), que repetida faz com que o sistema não consiga processar todas as informações no tempo definido.
- B) A rede é não reiniciável, então 2 casos podem se apresentar:
 - 1) A rede está com impasse, ou seja, existe no mínimo um estado morto na rede, fazendo com que nenhuma evolução na rede seja possível a partir deste estado.
 - 2) Existe mais de uma componente fortemente conexa no grafo de estados e no mínimo uma delas é uma componente fortemente conexa isolada (CFCI). Neste caso o sistema pode cair num loop improdutivo de forma que, embora a cada instante exista algum processamento sendo feito, o sistema não atinge o seu objetivo final.
- C) A rede é não viva, então 2 casos podem se apresentar:
 - 1) Pelo menos uma transição nunca é disparada, ou seja, algum evento do sistema jamais ocorrerá, pois as condições para tal nunca estarão satisfeitas.
 - 2) Existe mais de uma componente fortemente conexa no grafo de estados e no mínimo uma delas é uma CFCI para a qual nem todas as transições disparam. Neste caso o sistema pode cair num loop onde algumas ações da rede não serão executadas.

Em ambos os casos acima, indicações das eventuais imperfeições podem ser fornecidas pelo ANPRO: lugares não limitados, transições não disparáveis, estados não reiniciáveis e as CFCI existentes.

A fim de auxiliar o usuário na análise das causas dos diversos mau funcionamentos (que podem ser devidos a erro de concepção ou erros introduzidos na fase de modelamento da rede); a opção matriz de estados antecedentes para um dado estado do subprograma DOCRPN fornece para um dado estado o seu ramo ascendente que inclui os estados antecedentes e a seqüência de disparo.

Fornecendo o estado que apresenta o problema (estado superior, estado não reiniciável, etc.) obtemos os possíveis caminhos que a partir de E₀ conduzam a tal situação.

Assim com essas opções e mais algum conhecimento sobre o protocolo em estudo consegue-se aplicar a metodologia de análise para a sua validação.

5. EXEMPLO: PROTOCOLO DE TRANSFERÊNCIA DE DADOS: BIT ALTERNANTE

Como exemplo de protocolo para modelamento pela RPN e análise pelo ANPRO escolhemos o protocolo de Transferência de Dados Bit Alternante [1].

Este protocolo consiste basicamente de duas partes: um transmissor (conectado via meio de transmissão ao receptor) e um receptor.

5.1. DESCRIÇÃO DO PROTOCOLO DE BIT ALTERNANTE

O protocolo de bit alternante é provido de dois caminhos (direções) de transferência de dados e é do tipo envia - recebe. Antes de emitir uma nova mensagem, o processo emissor espera a chegada do reconhecimento da mensagem que foi enviada, ou seja, o protocolo fornece um serviço de transferência de quadro simples (mensagem), onde o usuário deve esperar por um reconhecimento antes de enviar as próximas mensagens. A perda de uma mensagem é corrigida através de retransmissões.

Para se garantir que as duplicatas possam ser reconhecidas pelo receptor, as mensagens são numeradas a priori pelo transmissor, com número de seqüência módulo 2 e para todas mensagens (mens) recebidas pelo receptor, um reconhecimento (rec_mens) é enviado com o número da seqüência da mensagem recebida. Assim são enviadas alternadamente mensagens com o número da seqüência igual a 1, mensagem do tipo1, ou a 0, mensagem do tipo2. Para ser enviada uma após outra, o transmissor espera o reconhecimento do envio da mensagem do tipo1 ou do tipo2, no caso da última mensagem ter sido do tipo1 ou do tipo2, respectivamente.

5.2. RPN DO PROTOCOLO DE BIT ALTERNANTE

Os dois caminhos (direções) de informação do protocolo são independentes. Consideraremos somente um caminho de transferência de dados, entre o transmissor e o receptor, pois o caminho oposto teria uma RPN semelhante. Estudar o comportamento de uma delas é suficiente para uma análise de validação do protocolo. A Fig. 6 mostra o protocolo de bit alternante modelado em RPN.

O lado esquerdo desta figura com um lugar e sete transições representa o transmissor. O meio de transmissão (ou interface) é representado por dois lugares (L2 e L3) e quatro transições no centro dessa figura. O lado direito com um lugar e sete transições representa o receptor.

As tabelas 1, 2, 3 e 4 apresentam maiores detalhes sobre a RPN, com relação às transições, aos predicados e operações, aos tipos de senhas e aos lugares, respectivamente.

Baseados nestas tabelas e no texto sobre o protocolo de Bit alternante, passemos, então, à descrição da RPN da Fig. 6.

O transmissor pode executar as seguintes ações:

- . início da transmissão (transição T1);
- . recebimento de OK1 e envio de mensagem do tipo2 (transição T2);
- . recebimento de OK2 e envio de mensagem do tipo1 (transição T3);
- . retransmissão (transições T4 e T5);
- . envio de fim de transmissão após recebimento de OK1 (transição T6);
- . envio de fim de transmissão após recebimento de OK2 (transição T7).

O meio de transmissão apresenta duas ações: perda de mensagem e perda de reconhecimento; considerando tanto mensagem do tipo1 como do tipo2. Nesta rede, tais perdas são representadas por transições que não têm lugares de saída. Essas ações são representadas pelas transições T8 (PERD_MENS1), T9 (PERD_MENS2), T10 (PERD_REC1) e T11 (PERD_REC2), respectivamente.

O receptor pode executar as seguintes ações:

- . sincronização do receptor com o transmissor no início de uma transmissão (transição T12);
- . envio de OK1 e OK2 (transições T13 e T14, respectivamente);
- . sinalização de final de transmissão (transições T15 e T16);
- . detecção de mensagens repetidas do tipo1 e do tipo2 (transições T17 e T18, respectivamente).

Se o transmissor tiver no lugar L1 uma senha S1 (bit 1) [ou S2 - bit 0], a mensagem a ser enviada será do tipo1 [ou do tipo2]. Se o receptor (lugar L4) contiver uma senha S1 [ou S2] indica que ele espera que a próxima mensagem seja do tipo1 [ou do tipo2].

A senha S4 foi utilizada para representar o final de transmissão. Utilizamos também a senha S5 para o sincronismo do receptor com o transmissor no início de uma transmissão para que o início da troca de dados fosse bem caracterizado.

5.3. ANÁLISE DO EXEMPLO UTILIZANDO O ANPRO

Utilizando o ANPRO para análise do protocolo de Bit Alternante modelado em RPN como mostra a Fig. 6, obtivemos, que a rede é limitada, reiniciável e viva, logo esta versão do protocolo não apresenta imperfeições. Os resultados da análise das propriedades da RPN equivalentemente, documentados pelo sub-programa DOCRPN estão no Apêndice 1. Ao final dos resultados da análise colocamos a representação gráfica do grafo de estados gerado. O grafo de estados apresenta todas as seqüências de disparo admissíveis para esta rede. Portanto verificamos que todas as ações deste protocolo são realizadas satisfatoriamente.

6. CONCLUSÃO

O programa ANPRO foi projetado para auxiliar na análise de validação de protocolos de comunicação, podendo realizar tanto uma análise das propriedades dinâmicas da RPN, como também uma simulação de seqüências bem determinadas assumidas pelos estados dos diversos processos que constituem o protocolo.

Dentro da análise das propriedades da RPN, o programa gera resultados úteis à correção de possíveis falhas de projeto dos protocolos. Ou seja, ele fornece diagnósticos sobre a RPN, tais como as listas de: transições que não disparam, estados mortos, estados antecedentes a um determinado estado, componentes fortemente conexas (CFC's).

Lembremos que o ANPRO foi desenvolvido em PASCAL, em microcomputador do tipo PCaT, com alocação dinâmica de memória, com uma capacidade para RPN's de 300 lugares, 300 transições e gerando até 1000 estados. Visando a utilização do ANPRO para análise de protocolos maiores (redes equivalentes maiores) pretendemos fazer sua implementação num computador de grande porte tipo VAX, o que em grande parte já foi facilitado pela linguagem de alto nível escolhida e pela estrutura dada ao pacote.

O ANPRO possui capacidade para analisar RPN's com até 5 tipos de senhas, 8 tipos de predicados e 6 tipos de operações. Uma última sugestão é a inclusão de outros tipos de senhas, de predicados e de operações com finalidade de abrangermos protocolos maiores e mais complexos.

REFERÊNCIAS

- [1] Bilington, J.; Wheeler, G.R.; Wilbur-Ham, M.C.; "PROTEAN: A High-level Petri Net Tool for Specification and Verification of Communication Protocols", IEEE, Transactions on Software Engineering, vol. 14, N.º 3, March 1988.
- [2] Bochmann, G.V; Geesi, J.; "A unified Method for the Specification and Verification of Protocols", IFIP, 1977.
- [3] CCITT Sixth Plenary Assembly Orange Book, Vol. IV.4, Recomendações Z101 a Z103, Genebra, 1986.
- [4] Damasceno, B.C.; Borelli, W.C; "Rede de Petri Numérica para Modelamento e Análise de Protocolos de Comunicação", Relatório Técnico DILIA, RT 54, Junho 1988.
- [5] Damasceno, B.C.; Modelos Baseados em Extensões de Rede de Petri para Análise de Protocolos de Comunicação, "Tese de Mestrado - FEE, UNICAMP, Junho 1989.
- [6] Diaz, M.; "Modelling and Analysis of Communication and Cooperation Protocols using Petri Net Based Models", North-Holland Publishing Company - Computer Networks 6 (1982) - pg. 419 - 441.
- [7] Merlin, P.M.; Farber, D.J.; "Recoverability of Communication Protocols Implications of heoretical Study", IEEE Transactions on Communications, Vol. Com. 24, N.º 9, pg. 1036 - 1043, September 1976.
- [8] Peterson, J.L.; "Petri Net Theory and Modelling of Systems", Prentice-Hall, Inc., 1981.
- [9] Symons, F.J.W.; "Introduction to Numerical Petri Nets, A General Graphical Model Concurrent Processing Systems", Australian Telecommunications Research, Vol. 14, N.º 1, 1980.

NOME	SIGNIFICADO
T1	Envia 1. ^a mensagem (ENV_MENS)
T2	Recebimento de OK1 (REC_OK1)
T3	Recebimento de OK2 (REC_OK2)
T4	Retransmissão 1 (Retrans1)
T5	Retransmissão 2 (Retrans2)
T6	Fim de transmissão após recebimento de OK1 (FIM1)
T7	Fim de transmissão após recebimento de OK2 (FIM2)
T8	Perda de mensagem1 (PERD_MENS1)
T9	Perda de mensagem2 (PERD_MENS2)
T10	Perda de reconhec. 1 (PERD_REC1)
T11	Perda de reconhec. 2 (PERD_REC2)
T12	1. ^a mensagem reconhecida (REC1)
T13	Envio de OK1 (ENV_OK1)
T14	Envio de OK2 (ENV_OK2)
T15	Recebe FIM2 (REC_FIM2)
T16	Recebe FIM1 (REC_FIM1)
T17	Deteção de mensagem repetida 1 (REP1)
T18	Deteção de mensagem repetida 2 (REP2)

TABELA 1 - TRANSIÇÕES DA RPN DA FIG. 6

TRANSIÇÃO	PREDICADOS (P)	OPERAÇÕES (OP)
T1	X1=0 / T1=OFF	T2=ON / T1=ON
T2		T2=ON
T3		T2=ON
T4	T2=OFF / T1=ON	T2=ON
T5	T2=OFF / T1=ON	T2=ON
T6		X1=1 / T1=OFF
T7		X1=1 / T1=OFF
T8		T2=OFF
T9		T2=OFF
T10		T2=OFF
T11		T2=OFF
T12		
T13		
T14		
T15		X1=0
T16		X1=0
T17		
T18		

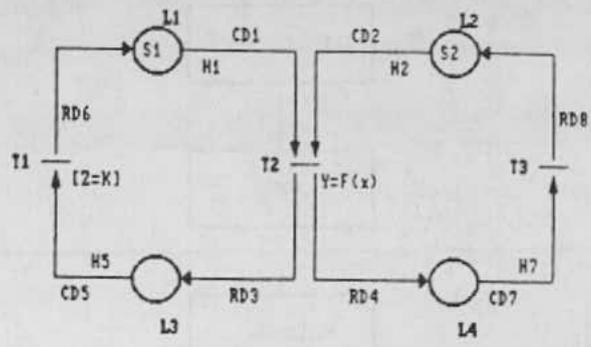
TABELA 2 - PREDICADOS E OPERAÇÕES DAS TRANSIÇÕES DA RPN DA FIG. 6

NOME		SIGNIFICADO
S	S1	BIT 1
E	S2	BIT 0
N	S3	MENSAGEM (QUADRO SIMPLES)
H	S4	FIM DE TRANSMISSÃO
A	S5	ESPERA DA 1. ^a MENSAGEM
S		(sinal de sincronismo)

TABELA 3 - TIPOS DE SENHAS DA FIG. 6

NOME		SIGNIFICADO
L U G A R E S	L1	Transmissor Livre
	L2	Meio (ou interface) entre Transm./Recep. (para envio de mensagem pelo transm.)
	L3	Meio (ou interface) entre Recep./Transm. (para envio de reconhecimento pelo receptor)
	L4	Receptor Livre

TABELA 4 - LUGARES DA RPN DA FIG. 6



ONDE:

-  : lugar
- L1,L2,L3,L4 : nomes dos lugares
- : transicao
- T1 : nome da transicao
-  : arco dirigido
- S1,S2 : tipos de senhas
- H1 : condicoes de habilitacao
- CD1 : condicoes de disparo
- RD1 : regras de pos-disparo
- $Y = F(x)$: operacao da transicao T2 (acao)
- $[Z = K]$: condicao de habilitacao de T1(predicado)
- X,Y,X,Z : variaveis da rede

FIG. 1: Exemplo de RPN

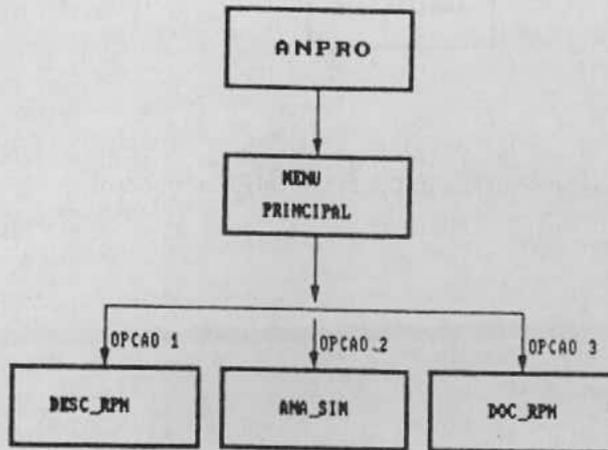


FIG. 2: Diagrama Modular do Programa Principal ANPRO

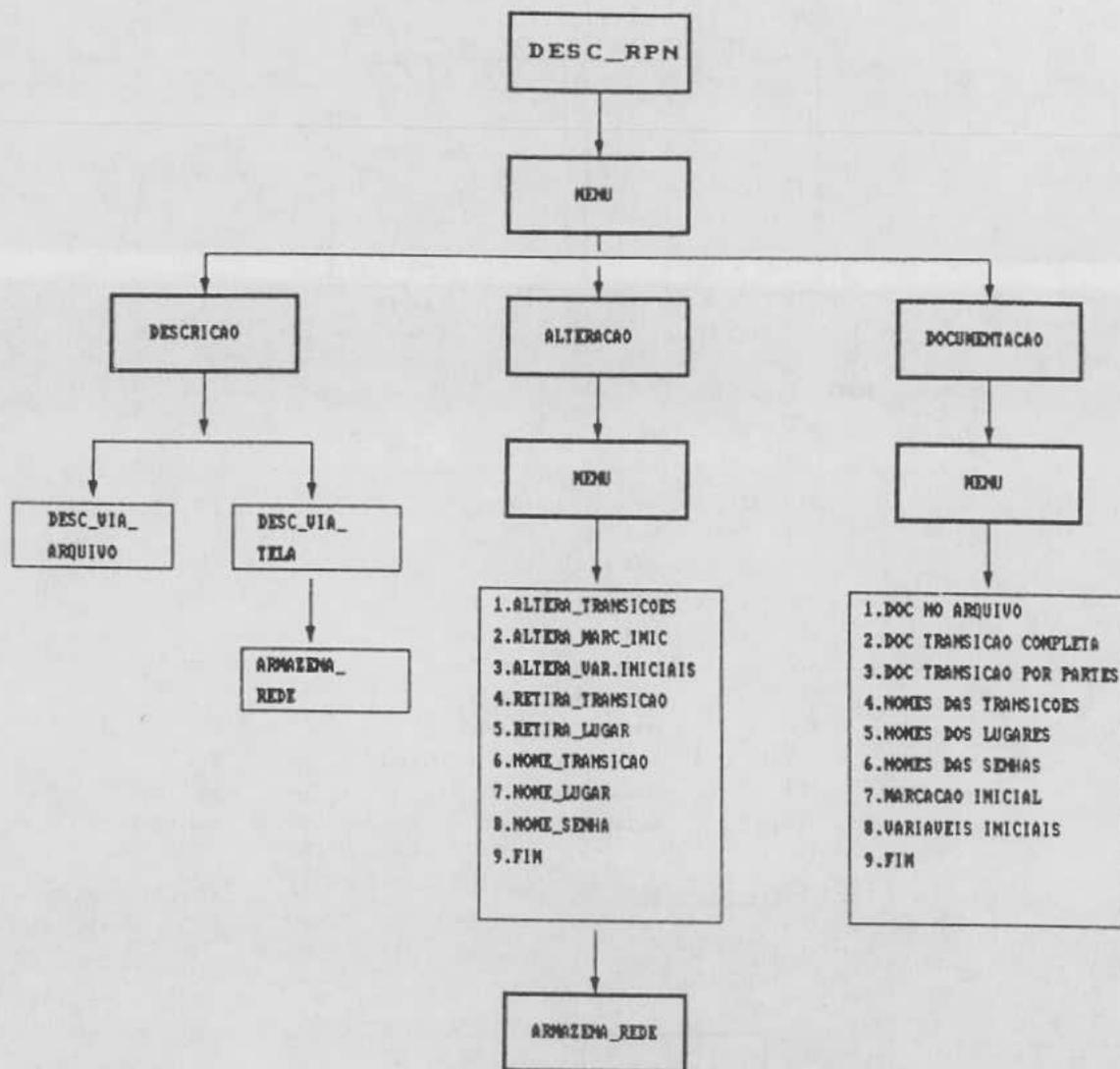


FIG. 3: Diagrama Modular da Opção 1 do Programa Principal

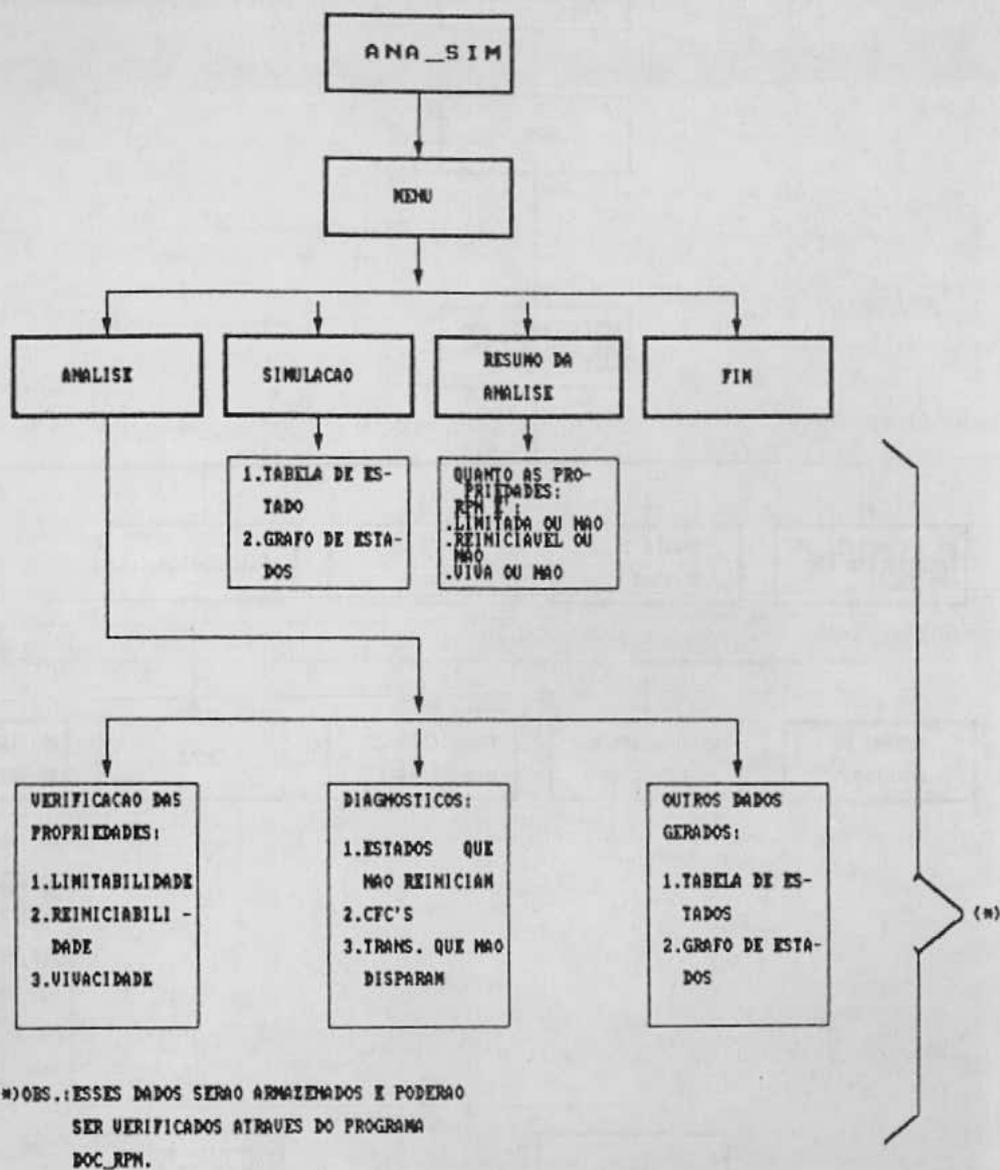


FIG. 4: Diagrama Modular da Opção 2 do Programa Principal

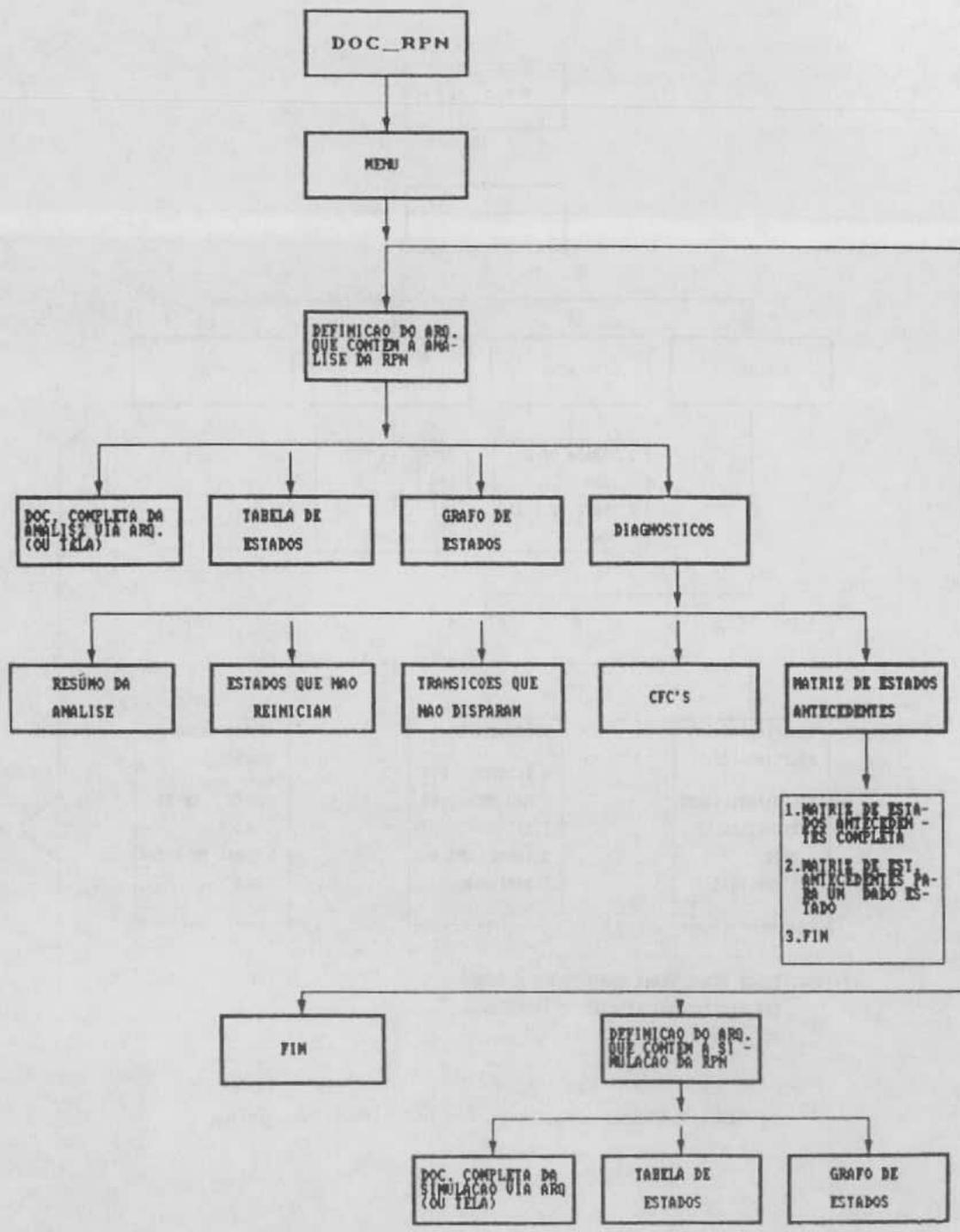


FIG. 5: Diagrama Modular da Opção 3 do Programa Principal

VARIÁVELS INICIAIS	VALORES
TEMPORIZADOR T1:	0
TEMPORIZADOR T2:	OFF

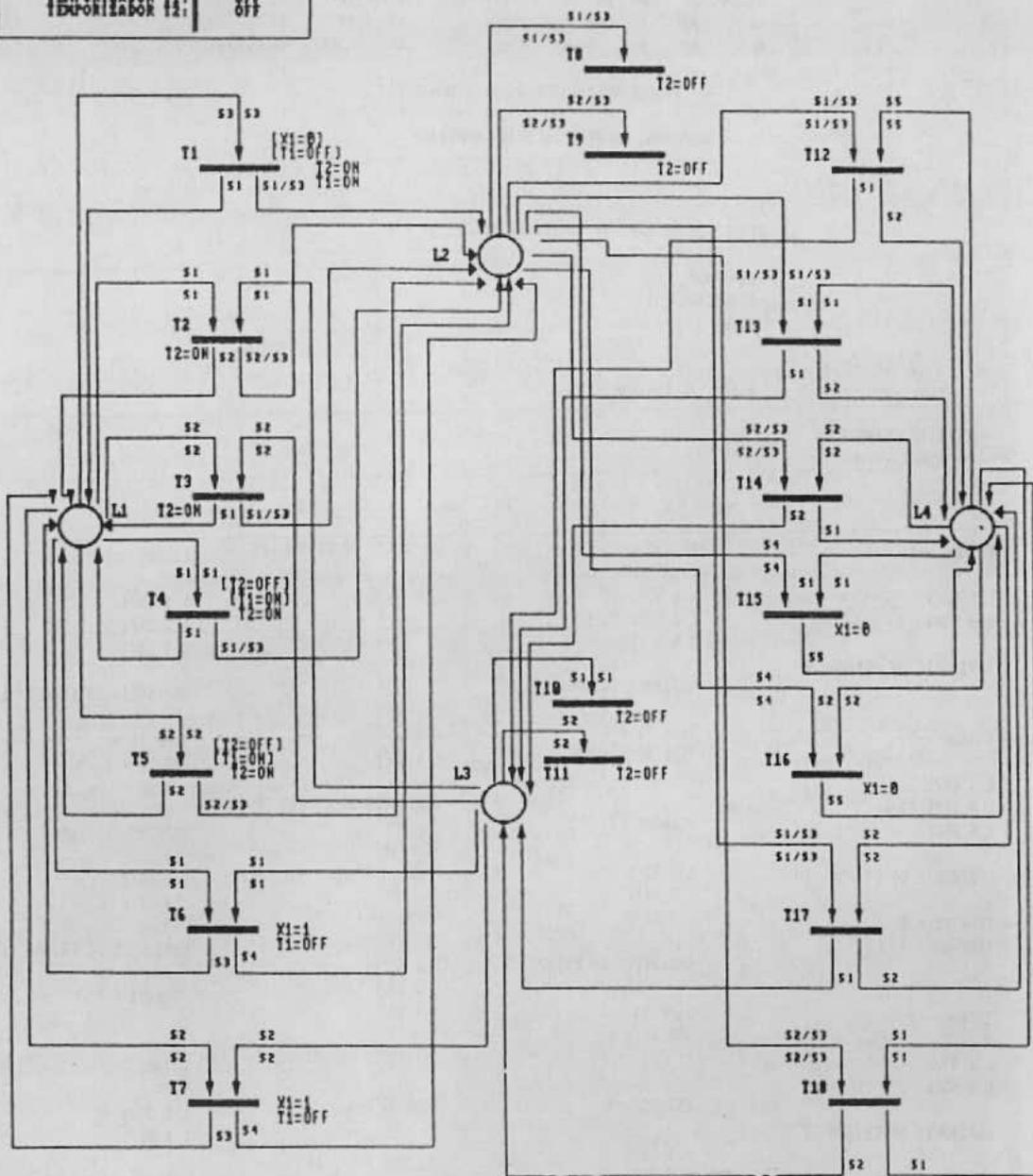


FIG. 6: RPN do Protocolo "Bit Alternante"

```

AAAAAAAAA  MNHN  MN  PFFFFFFF  RRRRRRRR  00000000
AA  AA  MN MN  MN  PF  PP  RR  RR  00  00
AAAAAAAAA  MN  MN  MN  PFFFFFFF  RRRRRRRR  00  00
AA  AA  MN  MN MN  PP  RR  RR  00  00
M:  MA  MN  MNHN  PP  RR  RR  00000000

```

ANALISADOR DE PROTOCOLOS DE COMUNICACAO

 MODELADOS EM REDE DE PETRI NUMERICA

RESULTADO DA ANLISE DAS PROPRIEDADES DA RPM

LIMITADA
 REINICIÁVEL
 VIVA

QTADE. DE ESTADOS = 15

TABELA DE ESTADOS

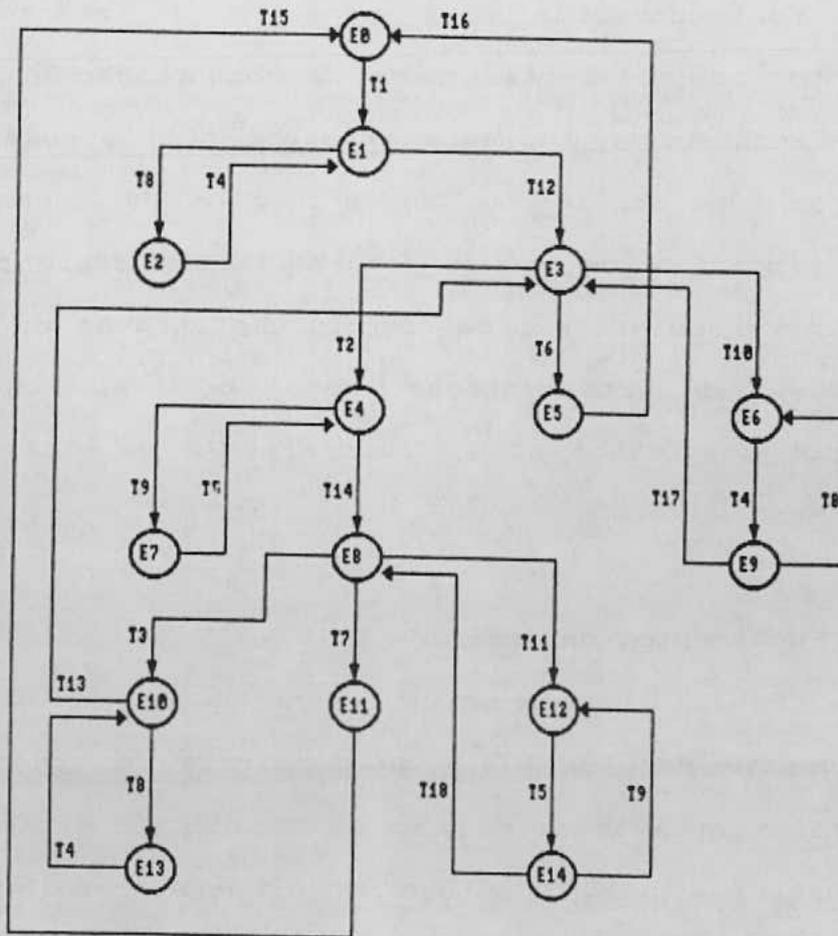
ESTADO --> 0	ESTADO --> 4	VARIAVEIS DO ESTADO 7	ESTADO --> 11
L 1 S3/1	L 1 S2/1	TEMP T1 = 1	L 1 S3/1
L 4 S5/1	L 2 S2/1 S3/1		L 2 S4/1
VARIAVEIS DO ESTADO 0	L 4 S2/1	ESTADO --> 8	L 4 S1/1
ESTADO --> 1	VARIAVEIS DO ESTADO 4	L 1 S2/1	VARIAVEIS DO ESTADO 11
L 1 S1/1	TEMP T1 = 1	L 3 S2/1	TEMP T1 = 1
L 2 S1/1 S3/1	TEMP T2 = 1	L 4 S1/1	VAR X1 = 1
L 4 S5/1	ESTADO --> 5	VARIAVEIS DO ESTADO 8	ESTADO --> 12
VARIAVEIS DO ESTADO 1	L 1 S3/1	TEMP T1 = 1	L 1 S2/1
TEMP T1 = 1	L 2 S4/1	ESTADO --> 9	L 4 S1/1
TEMP T2 = 1	L 4 S2/1	VARIAVEIS DO ESTADO 9	VARIAVEIS DO ESTADO 12
ESTADO --> 2	VARIAVEIS DO ESTADO 5	L 1 S1/1	TEMP T1 = 1
L 1 S1/1	TEMP T1 = 1	L 2 S1/1 S3/1	ESTADO --> 13
L 4 S5/1	VAR X1 = 1	L 4 S2/1	ESTADO --> 13
VARIAVEIS DO ESTADO 2	ESTADO --> 6	VARIAVEIS DO ESTADO 9	ESTADO --> 13
TEMP T1 = 1	L 1 S1/1	TEMP T1 = 1	L 1 S1/1
ESTADO --> 3	L 4 S2/1	ESTADO --> 10	L 4 S1/1
L 1 S1/1	VARIAVEIS DO ESTADO 6	L 1 S1/1	VARIAVEIS DO ESTADO 13
L 3 S1/1	TEMP T1 = 1	L 2 S1/1 S3/1	TEMP T1 = 1
L 4 S2/1	ESTADO --> 7	L 4 S1/1	ESTADO --> 14
VARIAVEIS DO ESTADO 3	L 1 S2/1	VARIAVEIS DO ESTADO 10	ESTADO --> 14
TEMP T1 = 1	L 4 S2/1	TEMP T1 = 1	L 1 S2/1
		TEMP T2 = 1	L 2 S2/1 S3/1
			L 4 S1/1
			VARIAVEIS DO ESTADO 14
			TEMP T1 = 1
			TEMP T2 = 1

APENDICE 1: RESULTADOS DA ANÁLISE PELO ANFPO DA PPN DA FIG. 6

GRAFO DE ESTADOS

~~E 0 - T1 -> E 1~~
~~E 1 - T2 -> E 2 - T8 -> E 2~~
~~E 2 - T4 -> E 1~~
 E 3 - T10 -> E 6 - T6 -> E 5 - T2 -> E 4
 E 4 - T9 -> E 7 - T14 -> E 8
 E 5 - T16 -> E 0
 E 6 - T4 -> E 9
 E 7 - T5 -> E 4
 E 8 - T3 -> E 10 - T7 -> E 11 - T11 -> E 12
 E 9 - T17 -> E 3 - T8 -> E 6
 E 10 - T13 -> E 3 - T8 -> E 13
 E 11 - T15 -> E 0
 E 12 - T5 -> E 14
 E 13 - T4 -> E 10
 E 14 - T18 -> E 8 - T9 -> E 12

NÃO HA CFC NESTA REDE



REPRESENTAÇÃO GRÁFICA DA TABELA DE GRAFO DE ESTADOS