

4º SBRC

RECIFE - 24 A 26 DE MARÇO 86

UM PACOTE DE SIMULAÇÃO PARA AVALIAÇÃO DE DESEMPENHO DE REDES LOCAIS

ANTONIO CAUPER FILHO

INSTITUTO DE TECNOLOGIA DA AMAZÔNIA
DEPARTAMENTO BÁSICO
69.000 - MANAUS - AM

JACQUES PHILIPPE SAUVÉ
JOSÉ ANTÃO BELTRÃO MOURA

UNIVERSIDADE FEDERAL DA PARAÍBA
DEPARTAMENTO DE SISTEMAS E COMPUTAÇÃO
GRUPO DE REDES DE COMPUTADORES
58.100 - CAMPINA GRANDE - PB

RESUMO: Este trabalho trata da concepção e implementação de um Pacote de Simulação para Avaliação de Desempenho de Redes Locais a partir de um Modelo Genérico de uma Rede Local.

1. INTRODUÇÃO

Redes Locais são redes para comunicação de computadores instalados em áreas geograficamente restritas, como por exemplo: um Campus Universitário. A sua organização estabelece a ligação de vários usuários à sub-rede de comunicação através de interfaces (veja Figura 1).

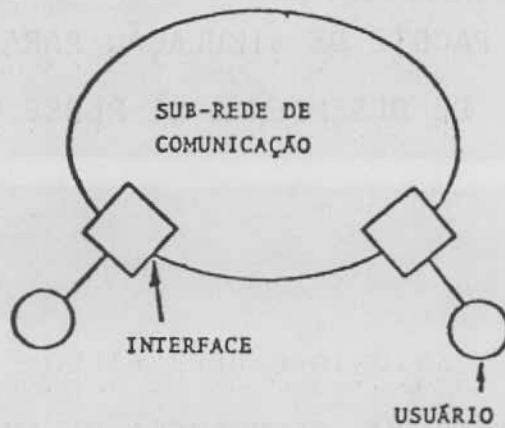


FIGURA 1: Organização da Sub-Rede de uma Rede Local.

A sub-rede de comunicação é composta das interfaces e um único meio de transmissão que é o responsável pelas transferências de informação entre as interfaces. O meio de transmissão é comumente disposto numa topologia em Anel (veja Figura 2) ou em Barra (veja Figura 3).

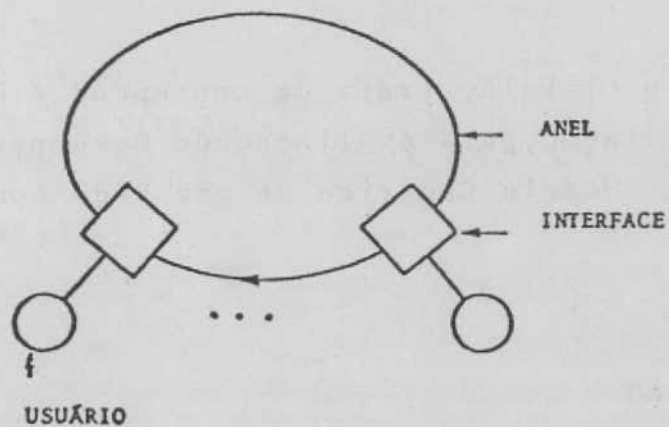


FIGURA 2: Anel.

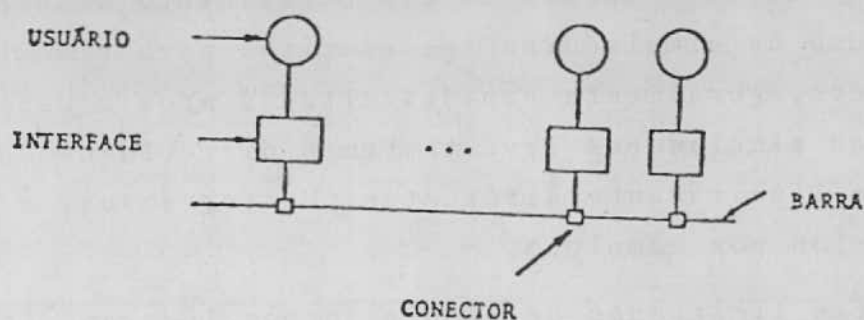


FIGURA 3: Barra.

A fim de ordenar o uso do meio de transmissão pelas várias interfaces na sub-rede, estas interfaces seguem as regras do chamado "protocolo de acesso". Em cima do protocolo de acesso, definem-se outros protocolos que possibilitam a comunicação fim-a-fim, entre usuários da Rede Local e que, viabilizam as próprias aplicações desses usuários. É de interesse, portanto, avaliar o desempenho dos protocolos de Redes Locais para que se possa determinar que arquitetura se presta melhor para o compartilhamento de recursos desejado.

Muito se tem feito em avaliação de desempenho de Redes Locais. Em [MOUR 83] é feito um estudo do desempenho da sub-rede de comunicação em Anel e em Barra através de Simulação Digital. Estudos analíticos de sub-redes locais podem ser encontrados em [BUX 81, MOUR 83/84]. Em [MOUR 85] é estudado o desempenho de protocolos de transporte. Em [WONG 83] é mostrado uma maneira analítica de se estudar o desempenho de Redes Locais a nível de aplicação.

Todos os estudos disponíveis baseiam-se em duas importantes técnicas para Avaliação de Desempenho: a Simulação e as Ferramentas Analíticas.

As Ferramentas Analíticas baseiam-se em Processos Estocásticos e Teoria das Filas. O problema existente nesta técnica é que é muito difícil de ser manuseada, e para simplificação dos modelos em estudo, algumas hipóteses são feitas, diminuindo assim o poder "prático" da ferramenta.

. A Simulação, porém, é bem mais simples e sempre funciona, servindo também para validar aproximações de Ferramentas Analíticas. Mas é cara em termos de desenvolvimento e de execução. Além disso, como os simuladores são escritos para resolverem problemas específicos, geralmente são descartados após o uso. A construção "ad hoc" de simuladores leva também à dificuldade de reusá-los em problemas ligeiramente diferentes. Nestes casos, é necessário desenvolvê-los por completo.

Estas limitações de Simulação nos levaram à necessidade de desenvolvimento de um Pacote Geral de Simulação de Redes Locais, que tivesse as seguintes características:

- a) Fácil interação com o usuário;
- b) Fácil integração com várias alternativas para diferentes níveis de protocolos da rede;
- c) Modularidade do Pacote;
- d) Modelar a Camada de Enlace do IEEE 802;
- e) Modelar a Sub-Camada de Acesso do IEEE 802:
 - Anel com passagem de Ficha;
 - Barra com passagem de Ficha;
 - Barra com CSMA-CD;
- f) Geração de Tráfego de acordo com a Aplicação;
- g) Interface com o usuário do pacote através de uma Linguagem de Entrada;
- h) Possibilidade de incluir detalhes do Protocolo de Transporte.

A contribuição deste trabalho vem da importância que ele tem para o usuário interessado em medir o desempenho de Redes Locais e principalmente para o projetista de Redes Locais, pois reduz o tempo de desenvolvimento de ferramentas para avaliação de desempenho e seu uso implica em custos apenas de execução.

É importante notar que nosso objetivo não é avaliar o desempenho de Redes Locais já existentes (isto já foi feito), mas sim, apresentar uma ferramenta útil para a implementação e alteração de Redes Locais.

Para que implementássemos o pacote com as características apresentadas anteriormente, foi necessário pensarmos em um Modelo Genérico para uma Rede Local que descrevemos a seguir.

2. MODELO GENÉRICO PARA UMA REDE LOCAL

2.1. Introdução

A arquitetura de uma Rede Local é geralmente composta das seguintes partes principais (do ponto de vista de avaliação de desempenho):

- a) Meio de Transmissão;
- b) Acesso ao Meio;
- c) Estação (ou Interface);
- d) Controle de Fluxo entre Estações;
- e) Controle de Fluxo para Acesso à Estação;
- f) Aplicação.

A Figura 4 ilustra um modelo (genérico) que captura detalhes das partes listadas acima.

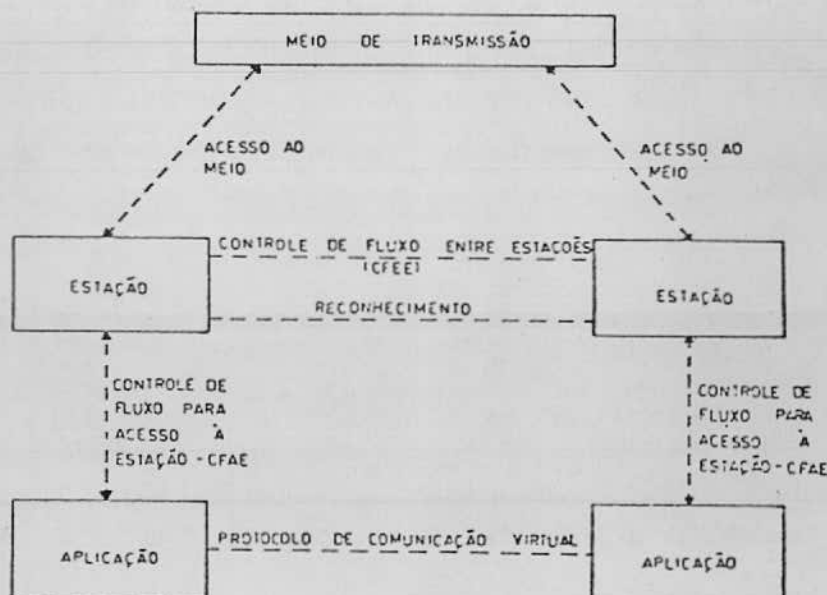


FIGURA 4: Modelo genérico para uma Rede Local.

2.2. Descrição das Entidades e suas Funções

Entidade é uma parte do Modelo que tem uma função própria (específica) e bem definida. Pode ser física, por exemplo: *Meio de Transmissão*; ou lógica, por exemplo: *Aplicação*. O modelo ilustrado na Figura 4 corresponde uma *Entidade* a cada uma das partes listadas na sub-seção 2.1.

Passaremos à descrição das Entidades do Modelo:

a) *Meio de Transmissão*

O *Meio de Transmissão* tem a função de transmitir as informações de uma Estação Fonte a uma Estação Destino. Para simplificar, no nosso Modelo o *Meio de Transmissão* será modelado como a Transmissão propriamente dita que por sua vez corresponde a um intervalo de tempo tirado de uma distribuição de probabilidade acrescido de um overhead.

b) *Acesso ao Meio*

Tem a função de controlar o acesso ao *Meio de Transmissão*, decidindo qual interface, em dado instante, tem permissão para transmitir. Podemos citar como exemplo o modelo do Acesso ao Meio de Transmissão em Anel com Passagem de Ficha. Neste protocolo uma ficha de controle (representada por uma seqüência especial de bits) é passada seqüencialmente ao redor do Anel. Qualquer interface, ao receber a ficha, pode removê-la do Anel, transmitir um pacote e então passar a ficha para a próxima interface.

c) *Estação*

Tem a função de transmitir e receber dados.

Para uma melhor distribuição modular do nosso Modelo, desmembramos a *Estação* em duas Entidades, que são:

- 1) *Processamento de Transmissão de Pacotes;*
- 2) *Processamento de Recebimento de Pacotes.*

O *Processamento de Transmissão de Pacotes* é modelado por uma Entidade que aceita pacotes (da *Aplicação*) a transmitir e para cada pacote aceito emite um aviso de aceitação deste pacote para a *Aplicação*.

O *Processamento de Recebimento de Pacotes* é modelado por uma Entidade que, para cada pacote de dados recebido, gera um pacote de reconhecimento (pacote ACK) a ser transmitido, se assim o usuário desejar. Se encarrega, ainda, de passar o pacote de dados que recebe para a Aplicação solicitada pelo usuário fonte, e de tomar as providências necessárias para o caso em que o pacote recebido é ACK.

d) *Controle de Fluxo entre Estações*

Tem a função de gerir os pacotes que serão transmitidos, mantendo uma fila, em cada Estação, destes pacotes e atualizando os recursos de Controle de Fluxo utilizados. É modelado como uma Entidade que aceita pedidos de transmissão e, dependendo das condições estabelecidas pelo Controle de Fluxo (ex.: tamanho da janela, créditos, etc.), libera uma autorização para (tentar) transmitir. Por exemplo, no Controle de Fluxo por janela, a Estação autoriza transmissões enquanto a janela estiver aberta. Quando a janela fechar, a Estação só dará uma nova autorização para transmitir quando receber um pacote de reconhecimento para algum pacote de dados previamente enviado.

e) *Controle de Fluxo para Acesso à Estação*

Controla os pacotes oriundos da Aplicação para acesso à Estação, mantendo uma fila destes pacotes e atualizando os recursos utilizados. É modelado como uma Entidade que aceita pacotes a transmitir em função das condições que a Estação estabelece para gerir estes pacotes (por exemplo: tamanho da fila). Ao aceitar um pacote, a Entidade sinaliza a Aplicação, informando o aceite.

f) *Aplicação*

Tem a função de interagir com o usuário do Pacote de Simulação, através da Linguagem de Entrada. É definida pelo usuário e modelada como uma Entidade que gera pacotes a transmitir de acordo com uma distribuição de probabilidade. Para cada pacote de dados gerado recebe a mensagem de aceitação deste pacote. (Vide Figura 5, a seguir).

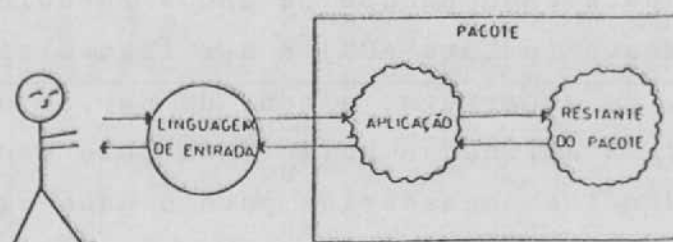
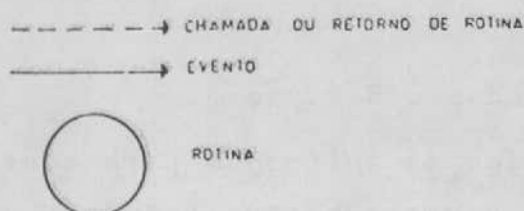


FIGURA 5: A aplicação e sua interação com o usuário e o restante do Pacote.

2.3. Relações entre as Entidades e seus Eventos

Vejamos a seguir como as Entidades do Modelo Genérico se relacionam através de Eventos e Chamadas a funções (rotinas). Por brevidade, nem todas as rotinas do pacote serão mencionadas aqui. Discutiremos apenas aquelas que forem chamadas por alguma Entidade ou acionadas por algum Evento.

Nas sub-seções a seguir usaremos a seguinte convenção:



Passaremos à descrição das Entidades e suas relações entre Eventos:

a) Entidade *Meio de Transmissão*

Esta entidade é acionada pelo Evento GANHA_ACESSO (escalonado pela Entidade *Acesso ao Meio*), que por sua vez chama a rotina DUPLICA_PACOTE. Esta rotina duplica o pacote a transmitir (ou melhor, a estrutura que representa o pacote) e em seguida escala o Evento REC_PAC, associando a ele o pacote duplicado, e escala ainda o Evento FIM_DE_TRANSMISSÃO, conforme mostra a Figura 6. Os Eventos REC_PAC e FIM_DE_TRANSMISSÃO são escalonados para ocorrerem simultaneamente, só que o primeiro é escalonado para a Estação Remota e o segundo para a Estação Fonte. O Evento REC_PAC

aciona a Entidade *Processamento de Recebimento de Pacotes Remota* e o Evento *FIM_DE_TRANSMISSÃO*, a Entidade *Acesso ao Meio Fonte*.

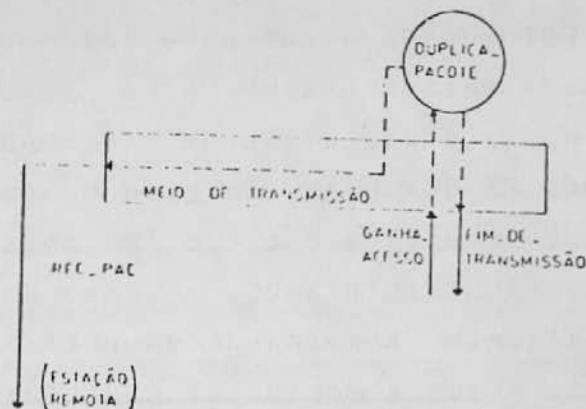


FIGURA 6: Modelo da Entidade Meio de Transmissão.

b) Entidade *Acesso ao Meio*

Nós dividimos esta Entidade em duas partes: a parte Independente do Protocolo de Acesso ao Meio e a Dependente. Isto foi feito para não quebrar a modularidade do modelo, visto que as ações e procedimentos do Acesso ao Meio são diferentes para diferentes Protocolos de Acesso ao Meio. Por exemplo: as ações requisitadas por um Protocolo de Passagem de Ficha são diferentes das requisitadas por um Protocolo com Inserção de Registro. Assim, se o usuário quiser um protocolo não implementado no Pacote basta mudar a parte Dependente. (Vide Figura 7).

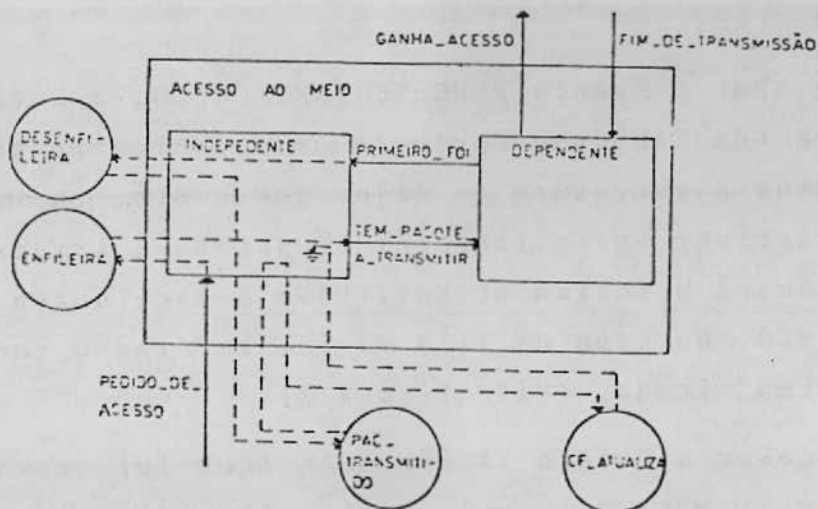


FIGURA 7: Modelo da Entidade Acesso ao Meio.

A parte Independente é acionada pelo Evento `PEDIDO_DE_ACESSO`, que chama a rotina `ENFILEIRA`, responsável por colocar em uma fila o pacote a transmitir, escalonando em seguida o Evento `TEM_PACOTE_A_TRANSMITIR`. A parte Independente é também acionada pelo Evento `PRIMEIRO_FOI` (que é escalonado pela parte Dependente) que por sua vez chama a rotina `DESENFILEIRA`. Esta rotina retira o pacote da fila e depois chama a rotina `PAC_TRANSMITIDO` que poderá escalonar o Evento `TX_SEM_CFAE`, no caso do pacote necessitar ser retransmitido (estourou time-out e o ACK correspondente não chegou), ou o Evento `PACOTE_LIBERADO`, no caso em que o pacote foi transmitido normalmente. Retornando da rotina `PAC_TRANSMITIDO`, a parte Independente chama a rotina `CF_ATUALIZA`, que se ocupa em atualizar os recursos de Controle de Fluxo utilizados pelo pacote ora transmitido. Em seguida, `CF_ATUALIZA` escalona o Evento `PODE_TENTAR_ACESSO`, caso o Controle de Fluxo for por crédito; porém, se for por janela, o Evento `PODE_TENTAR_ACESSO` só será escalonado com a chegada de um pacote ACK.

A parte Dependente é acionada pelo Evento `TEM_PACOTE_A_TRANSMITIR` que por sua vez escalona o Evento `GANHA_ACESSO` em função das condições estabelecidas pelo Protocolo de Acesso ao Meio. É acionada também pelo Evento `FIM_DE_TRANSMISSÃO` escalonando em seguida o Evento `PRIMEIRO_FOI`.

Vale salientar que a rotina `CF_ATUALIZA` faz parte também da Entidade *Controle de Fluxo entre Estações* (CFEE).

c) Entidade *Processamento de Transmissão de Pacotes*

Esta Entidade é acionada por dois Eventos: `PODE_TENTAR_ACESSO` e `TX_SEM_CFAE`.

Ao receber o Evento `PODE_TENTAR_ACESSO`, a Entidade chama a rotina `CF_OK` (da Entidade *Controle de Fluxo entre Estações*) que é quem examina a estrutura de dados que contém pacotes (fila). Se a fila não estiver vazia (isto é, há pacote a transmitir) a rotina `CF_OK` chamará a rotina `DESENFILEIRA` e escalonará o Evento `PE_DIDO_DE_ACESSO`. No caso de fila vazia, retorna o controle para a Entidade acima citada. (Vide Figura 8)

Ao receber o Evento `TX_SEM_CFAE` (que foi escalonado pela rotina `PAC_TRANSMITIDO` ou pela rotina `REC_PAC`) chama a rotina `EN`

FILEIRA para enfileirar o pacote a ser transmitido e em seguida chama a rotina CF_OK, a qual é avisada da chegada deste pacote e se tiver crédito ou janela disponível (enfim, recursos para transmissão), chama a rotina DESENFILEIRA e logo após, escalona o Evento PEDIDO-DE-ACESSO. (Veja Figura 8).

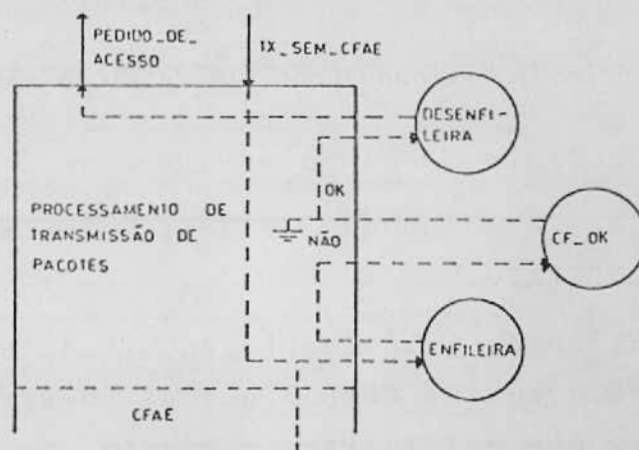


FIGURA 8: Modelo da Entidade Processamento de Transmissão de Pacotes.

Esta Entidade pode ainda ser acionada pela Entidade *Controle de Fluxo para Acesso à Estação*, através de chamada à rotina ENFILEIRA.

É importante salientar que esta Entidade é muito relacionada com a Entidade *Controle de Fluxo para Acesso à Estação*. Por isso, a Figura 8 mostra o CFAE como uma continuação da Entidade *Processamento de Transmissão de Pacotes*.

d) Entidade *Controle de Fluxo entre Estações*

Esta Entidade não é acionada por nenhum evento. São chamadas de rotinas que a ativam: a chamada da rotina CF_OK proveniente da rotina ENFILEIRA, outra chamada da mesma rotina proveniente da Entidade *Processamento de Transmissão de Pacotes*. É ativada também pela chamada da rotina CF_ATUALIZA que pode ser de dois modos: pela Entidade *Acesso ao Meio* e pela Entidade *Processamento de Recebimento de Pacotes*. A rotina CF_ATUALIZA escalona o Evento PODE_TENTAR_ACESSO. (Veja Figura 9).

Vale salientar que esta Entidade é opcional.

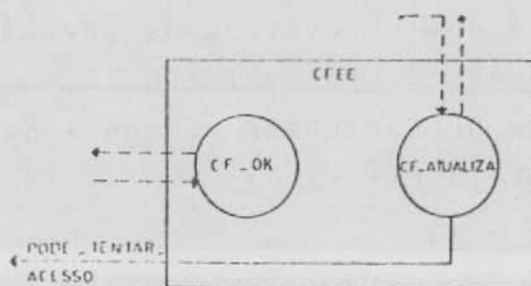


FIGURA 9: Modelo da Entidade Controle de Fluxo entre Estações.

e) Entidade *Controle de Fluxo para Acesso à Estação*

Esta Entidade é acionada por dois Eventos: `PACOTE_A_TRANSMITIR` e `PACOTE_LIBERADO`.

Quando é acionado pelo Evento `PACOTE_A_TRANSMITIR` (veja a Figura 10), chama a rotina `CFAE_B` que se encarrega de chamar a rotina `ENFILEIRA`, que enfileirá o pacote, na disponibilidade de vaga na fila. Neste caso, escalona o Evento `PACOTE_ACEITO`. Caso não haja condições de enfileirar, o pacote ficará bloqueado.

Quando é acionada pelo Evento `PACOTE_LIBERADO`, chama a rotina `CFAE_DB` que se encarrega de enfileirar o pacote que foi bloqueado anteriormente e, em seguida, escalona o Evento `PACOTE_ACEITO`. Caso não haja um pacote bloqueado não faz nada.

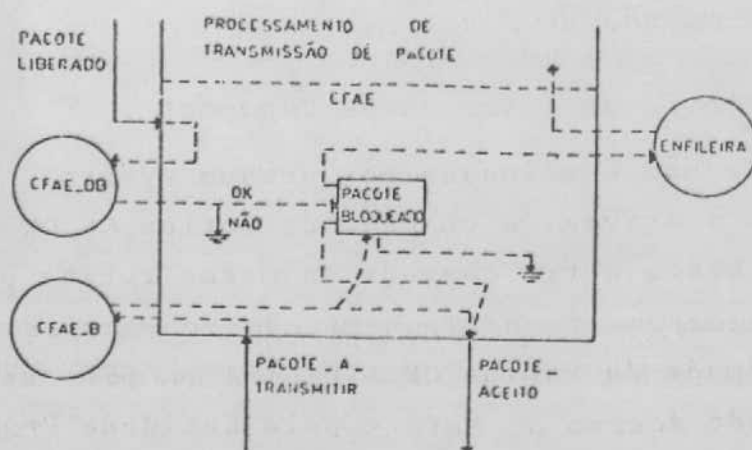


FIGURA 10: Modelo da Entidade Controle de Fluxo para Acesso à Estação.

O aparecimento da Entidade *Processamento de Transmissão de Pacotes* na Figura 10 é justificada no item c desta seção.

f) Entidade *Processo de Aplicação*

É acionada pelo Evento `PACOTE_ACEITO`, e em fase de inicialização pela aplicação escolhida e/ou definida pelo usuário.

Para cada Evento `PACOTE_ACEITO` recebido, um Evento `PACOTE_A_TRANSMITIR` é escalonado através de um gerador de tráfego escolhido pelo usuário (caso a opção de bloqueio da aplicação pelo Controle de Fluxo para Acesso à Estação estiver ativa). Se não for bloqueada, a Entidade *Aplicação* armazenará em uma fila os pacotes que serão submetidos à estação.

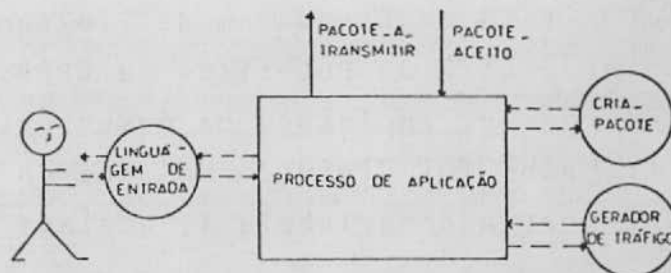


FIGURA 11: Modelo da Entidade Processo de Aplicação.

g) Entidade *Processamento de Recebimento de Pacotes*

É acionada pelo Evento `REC_PAC` (recepção de pacote).

Quando o pacote recebido é de dados e o usuário pediu inclusão de tráfego de reconhecimento (ACK), um pacote ACK é gerado e associado a um Evento `TX_SEM_CFAE`, que é escalonado de pronto. Em seguida, o pacote recebido é liberado. No caso em que o tráfego de ACKs é descartado, esta Entidade só fará a coleta de Estatísticas e a liberação do pacote recebido.

Quando o pacote recebido for ACK a Entidade chama a rotina `CF_ATUALIZA` que atualizará os recursos utilizados para Controle de Fluxo (por exemplo: abrir a janela). No caso de Controle de Fluxo por janela, escalona o Evento `PODE_TENTAR_ACESSO` e, em seguida, libera o pacote recebido.

Vale salientar que o tamanho do Pacote em bytes é variável devido à criação de Estruturas de Dados dinamicamente relocáveis, em fase de execução. O tamanho estático do Pacote (não em execução) é da ordem de 35 Kbytes.

A entrada dos dados é feita através de uma Linguagem de Entrada, bem simples e, ainda, sem muitos recursos.

Várias estatísticas podem ser extraídas do Pacote e é o usuário quem as escolhe do conjunto de estatísticas definidas e implementadas (não é difícil para o usuário incluir novas estatísticas). Algumas estatísticas que o Pacote pode fornecer são do tipo: histograma de uma variável à escolha do usuário, vazão média da sub-rede, desvio padrão de uma variável, média de uma variável, atraso médio e outras.

A Validação deste Pacote foi feita para o caso de uma Rede Local em Anel com Passagem de Ficha, comparando as características de atraso vs vazão obtidas com o Pacote e os resultados das referências [MOUR 83] e [MOUR 85].

4. CONCLUSÕES E SUGESTÕES PARA EXTENSÃO DO TRABALHO

Pensamos que este trabalho coloca à disposição de usuários e projetistas de Redes Locais uma ferramenta para a Avaliação de Desempenho dessas Redes, na forma de um simulador modular, cuja funcionalidade pode ser estendida com adição de novos blocos.

A incorporação de novos blocos é possibilitada pela normalização das interfaces (em termos de Eventos e Chamadas a rotinas) entre os vários módulos do Pacote.

Algumas extensões a este trabalho, são sugeridas na Tabela 2.

EXTENSÃO	ESTIMATIVA DE HOMENS/HORA NECESSÁRIAS
Acesso ao Meio - Parte Dependente: Barra com Passagem de Ficha	80
Acesso ao Meio - Parte Dependente: Barra com CSMA-CD	100
Controle de Fluxo entre Estações por Crédito	75
Processo de Aplicação - Transmissão de Voz e Dados	90
Processo de Aplicação - Transmissão de Arquivos	70

TABELA 2: Sugestões para Extensão do Trabalho.

AGRADECIMENTOS

Agradecemos o suporte do CNPq a este trabalho.

5. REFERÊNCIAS

- [BUX 81] Bux, W., "Local Área Subnetworks: A Performance Comparison", IEEE Trans. on Commun., Vol. COM-29, nº 10, Outubro, 1981.
- [CHAN 78] Chandy, K. Mani e Yek, Raymond T., "Current Trends in Programming Methodology", "Software Modeling", Vol. 3, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 07632, 1978.
- [GIOZ 85] Giozza, W.F., Araújo, J.F.M., Moura, J.A.B. e Sauvê, J.P., "Tendências de Padronização da Sub-Rede de Comunicação em Redes Locais de Computadores", GRC/UFPB, Relatório Técnico nº RT-02/85, Fevereiro, 1985.
- [IEEE 802] IEEE Study Group on Local Area Network; Project 802.
- [MOUR 83] Moura, J.A.B. e Sauvê, J.P., "Avaliação de Desempenho de Redes de Dados Locais", Anais do 1º SBRC, Porto Alegre, Maio, 1983.
- [MOUR 83/84] Moura, J.A.B. e Sauvê, J.P., "Análise Assintótica do Desempenho de Redes de Dados Locais", RBC, Rio de Janeiro, Vol. 3, nº 2, 1983/84.

- [MOUR 84] Moura, J.A.B., Sauvê, J.P., Giozza, W.F. e Araújo, J. F.M., "Técnicas de Simulação Digital", GRC/UFPB, Relatorio Técnico nº RT-18/84, Dezembro, 1984.
- [MOUR 85] Moura, J.A.B., Sauvê, J.P., Giozza, W.F. e Araújo, J. F.M., "Avaliação de Desempenho de Protocolos de Transorte", GRC/UFPB, Relatorio Técnico nº 09/85, 1985.
- [SAGE 78] Sager, G.R. e Wong, J.W., "A First Course in Simulaotion", CS-78-03 - Computer Science - University of Wateroloo. Waterloo, Ontário, Canadá, 1978.
- [TROP 81] Tropper, C., "Local Computer Network Technologies", Acaoademic Press, Inc., Nova Iorque, 1981.
- [WONG 83] Wong, J.W., Moura, J.A.B. e Field, J.A. - "Hierarchiocal Modelling of File Transfers on Local Area Networks", Comput. & Elect. Engng., Vol. 10, nº 3, 1983, pp. 191-207.