

PROJETO DE UM SUBSISTEMA DE REDES PARA SISTEMA  
OPERACIONAL COMPATIVEL COM UNIX (1.1).

Celso Gonzalez Hummel  
Ivan Patricio Miranda  
Mirian Favareto de Macedo  
Wang Kuei Yu

Laboratório de Sistemas Integráveis  
Departamento de Engenharia Elétrica  
Escola Politécnica da USP  
Av. Prof. Luciano Gualberto, tv.3, 158  
Cidade Universitária - São Paulo  
CEP 05508

Sumário: O trabalho apresenta algumas considerações sobre as características incorporadas pela AT&T no sistema operacional UNIX System V Release 3 para a implementação de facilidades de redes no contexto de desenvolvimento do projeto de redes no Laboratório de Sistemas Integráveis, e o desenvolvimento de um subsistema de comunicações, que atenda com eficiência uma rede local com o Minissupercomputador, estações de trabalho e super-microcomputadores, além de uma rede de teleprocessamento para um grande número de terminais.

## 1 - Introdução

A partir da implementação da rede local Ethernet, em 1973, em um equipamento da classe de estações de trabalho, grandes contribuições têm sido feitas para o setor de redes locais. Algumas das mais importantes que podemos citar, recentemente, são: a introdução, realizada em 1983, de recursos de rede em um ambiente UNIX(\*) na versão BSD 4.2 de Berkeley e em 1986 no UNIX(\*) System V, Release 3.

Múltiplas topologias e novas aplicações têm surgido no mercado de redes locais de alto desempenho. O mercado crescente de estações de trabalho em ambiente UNIX vem suportando novas aplicações baseadas em rede, como compartilhamento de um sistema de arquivos ou terminais remotos (de recursos gráficos e baseados em janelas, localizados através da rede) que exigem uma rede alta de velocidade e desempenho para operar adequadamente.

### 1.1 - O Projeto de Redes no LSI

Com a conclusão dos projetos Supermicrocomputadores (SP32-I e SP32-III) e Estações Gráficas (EG880X) no Laboratório de Sistemas Integráveis da Escola Politécnica da Universidade de São Paulo e com a crescente demanda de redes de comunicação, surgiu a necessidade de implementação de uma rede interligando as máquinas citadas e o Mini-supercomputador (em projeto), visando o aumento da eficiência, extensibilidade, disponibilidade e compartilhamento de recursos disponíveis.

As máquinas mencionadas acima têm em comum o sistema operacional desenvolvido no LSI, compatível com UNIX System V da AT&T. Dando continuidade aos projetos já realizados, é desejável que o sistema operacional forneça suporte para a implementação de serviços de protocolos de comunicação, visando minimizar o esforço de projeto de rede de computadores.

O sistema operacional UNIX foi, durante muito tempo, criticado pela ausência de um suporte consistente para implementação de serviços de rede de comunicação de dados. Disso resultaram muitas implementações particulares de protocolos e utilitários de redes sobre o tradicional subsistema de entrada/saída de caracteres. A diversificação das implementações tornou os sistemas incompatíveis.

A AT&T, proprietária do UNIX System V especificou e implementou recentemente uma versão do Sistema Operacional onde incorporou características novas, cujo principal objetivo é atender às necessidades atuais do mercado, fornecendo suporte à construção de redes de computadores nos sistemas UNIX.

Faremos aqui algumas considerações sobre estas características incorporadas pela AT&T dentro do contexto de desenvolvimento do projeto de redes no LSI, além do desenvolvimento de um subsistema de comunicações, que atenda com eficiência uma rede local, com o Minisuper, estações de trabalho e super-micros; e de uma rede de teleprocessamento para um grande número de terminais.

## 2 - Suporte do Sistema Operacional para Implementação de Redes de Computadores

### 2.1 - Suporte para Interconexão de Computadores

A construção de um conjunto mínimo de suporte adequado aos serviços de comunicação no Sistema UNIX deve cumprir os seguintes objetivos:

1. Fornecer uma interface padronizada para os serviços de rede - o sistema deve fornecer uma interface através da qual um programa aplicativo pode receber os serviços da rede, sem se preocupar em saber qual das famílias de protocolos está fornecendo estes serviços. Desta forma, garante-se a portabilidade dos aplicativos desenvolvidos.

2. Prover serviços usuais de rede de forma independente de protocolos - o sistema deve prover um conjunto mínimo de serviços de redes tais como:

- os tradicionais: correio eletrônico, transferência de arquivos, execução remota de tarefas;
- os aplicativos comerciais: videotexto e transferência eletrônica de fundos, e
- os serviços tradicionais do sistema operacional: gerenciamento de sistema de arquivos e de processos distribuídos através da rede.

Esses serviços devem ser oferecidos junto com o sistema na forma de protocolos de comunicação independentes, isto é, uma única implementação de cada serviço deve ser capaz de operar em qualquer ambiente de protocolos e meios físicos.

3. Fornecer ferramentas para a implementação de protocolos de maneira modular e reconfigurável - as vantagens de uma estrutura modular de protocolos (evidenciadas pelas modernas arquiteturas de protocolos e principalmente pelo Modelo de Referência OSI da ISO), incluem a capacidade de isolar um protocolo ou aplicativo de alto nível de protocolos de níveis mais baixos. Isto é, possibilitam a substituição dinâmica de um protocolo, como por exemplo o TCP (Transmission Control Protocol) por um outro protocolo funcionalmente equivalente, como o protocolo de transporte classe 4 da ISO.

As características mencionadas acima vêm de encontro a dois princípios básicos do Modelo OSI:

- particionamento funcional de protocolos de comunicação em camadas distintas, e
- definição de interfaces de serviços - cada camada prove serviços à camada imediatamente superior de forma que esta não precise preocupar-se em saber como os serviços são fornecidos.

## 2.2 - Visão Geral da Interface de Transporte do UNIX (TLI)

As sete camadas do Modelo OSI representam o "esqueleto" geral das arquiteturas modernas de redes de comunicação. Associada a cada camada do Modelo, existe uma definição de serviços que estabelece as obrigações de um protocolo naquela camada. Muitos protocolos são capazes de fornecê-los, mas os usuários possuem uma definição precisa (não ambígua) dos serviços que qualquer um dos protocolos possa oferecer.

Das sete camadas do Modelo OSI, destaca-se aqui a camada de transporte. Os protocolos de transporte provêm comunicação confiável entre máquinas independentemente da topologia sobre a qual a rede está construída e, mais importante ainda, a camada de transporte define um conjunto de serviços comuns às camadas de muitas séries modernas de protocolos. Exemplos desses protocolos são os protocolos da ISO (International Standard Organization), o protocolo TCP/IP (Transmission Control and Internet Protocol) da ARPANET, o protocolo XNS (Xerox Network Systems) e o protocolo SNA (Systems Network Architecture).

A solução para a construção de um ambiente que suporte a rede no sistema UNIX pode ser adequada dentro das definições de serviços do Modelo OSI. Definindo-se uma interface de biblioteca (Transport, Level Interface - TLI), possibilita-se a implementação de aplicativos e camadas superiores de rede sem a necessidade de conhecimento prévio

da topologia e características da rede abaixo desta interface como mostra a figura a seguir.

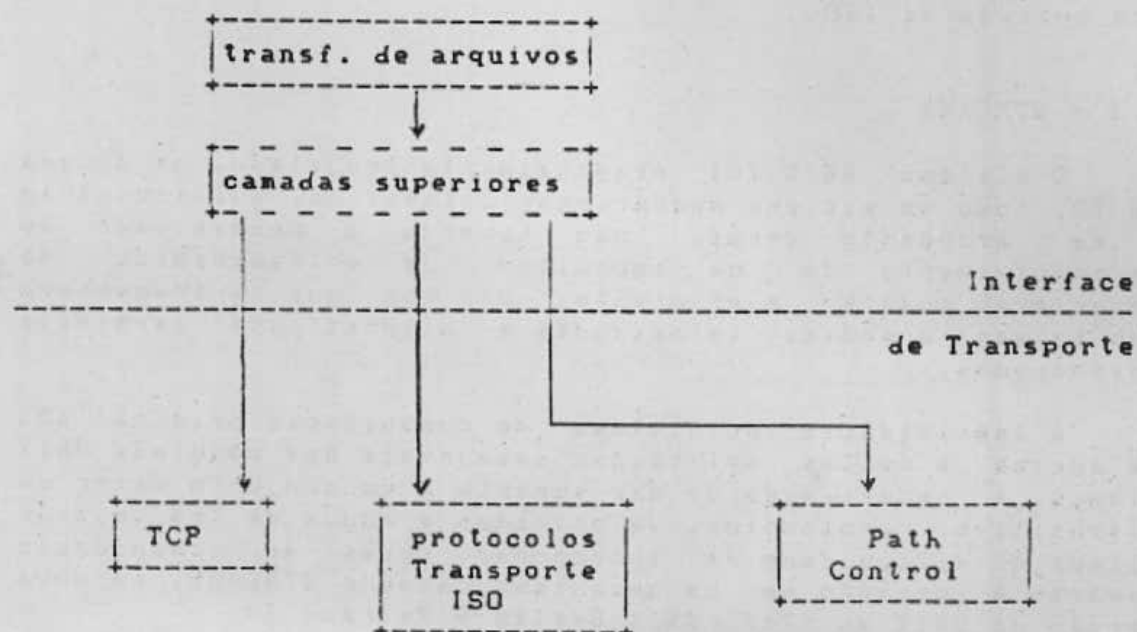


Fig. 1 - Independência de protocolos

A Interface de Transporte do sistema UNIX é baseada na definição de serviços de Transporte da ISO (ISO 8072) e é implementada como uma biblioteca de usuário usando um mecanismo de entrada/saída, STREAMS.

A Interface de Transporte oferece três grupos de serviços:

- primitivas de gerenciamento comum - provêem serviços de gerenciamento local. São utilizadas na fase onde um usuário de transporte estabelece, tanto utilizando o serviço no modo com conexão quanto no modo sem conexão, um ponto de acesso de transporte (transport access point), e associa um endereço de transporte a este ponto. Isto é, através desta interface, o aplicativo pode abrir o provedor de serviços de transporte, associando-se a um endereço local e negociar as opções de protocolos ou definir quaisquer condições de contorno.

- primitivas de serviços de transporte no modo com conexão - são primitivas fornecidas pelos protocolos que suportam serviço confiável de circuito virtual (TCP, SPP, OSI, Path Control) de modo que um aplicativo, através destas primitivas, pode abrir, enviar, receber dados e liberar um circuito virtual.



- primitivas de serviços de transporte no modo sem conexão - serviços fornecidos pelos protocolos que usam o serviço de datagrama (DARPA e protocolo de transporte modo sem conexão da ISO).

### 2.3 - STREAMS

O sistema UNIX foi originalmente projetado, na década de 70, como um sistema operacional interativo, multi-usuário e de propósito geral, não havendo a necessidade do desenvolvimento de um mecanismo de entrada/saída de caracteres modular e eficiente, uma vez que se trabalhava com baixas e médias velocidades e dispositivos terminais assíncronos.

A ineficiência do sistema de comunicação original não se adequa a muitas aplicações desejáveis nas máquinas UNIX atuais. A necessidade de dar suporte a um conjunto maior de dispositivos, protocolos, velocidades e modos de transmissão aliada à dificuldade de incorporar redes de comunicação levaram à criação de um mecanismo chamado STREAMS, na nova versão do UNIX da AT&T (UNIX System V Release 3).

STREAMS define interfaces padrões para entrada/saída de caracteres dentro do núcleo e entre o núcleo e o resto do sistema UNIX. Estas interfaces padrões juntamente com a simplicidade do mecanismo, permitem desenvolvimentos modulares e portáteis e facilitam a migração de serviços e componentes da rede com maior eficiência.

O mecanismo de STREAMS consiste de um conjunto de chamadas de sistema, rotinas utilitárias e recursos do núcleo do sistema operacional usados para criar, utilizar e finalizar um caminho de fluxo de dados bidirecional (full-duplex), o qual conecta um "driver" de algum dispositivo a um processo de usuário.

Um Stream pode ser visto como um canal de comunicação para processamento e transferência bidirecional de dados entre um "driver" de caracteres no núcleo e um processo de usuário. O "driver" no núcleo é a rotina que trata diretamente o "hardware" de comunicações.

Um Stream é composto das seguintes estruturas: "Stream head" (cabeça), módulo(s) (opcional) e um "driver":

- Stream head - provê uma interface entre o Stream e o processo de usuário. Sua principal função é processar as chamadas de sistema relacionadas ao mecanismo de STREAMS feitas pelo usuário.

- Módulo - é o "software" que trata os dados que fluem entre o "Stream head" e o "driver".

- Driver - um "driver", neste caso, pode ser de dispositivo, provendo serviços de um dispositivo externo de entrada/saída, ou de "software" interno (pseudo "device driver").

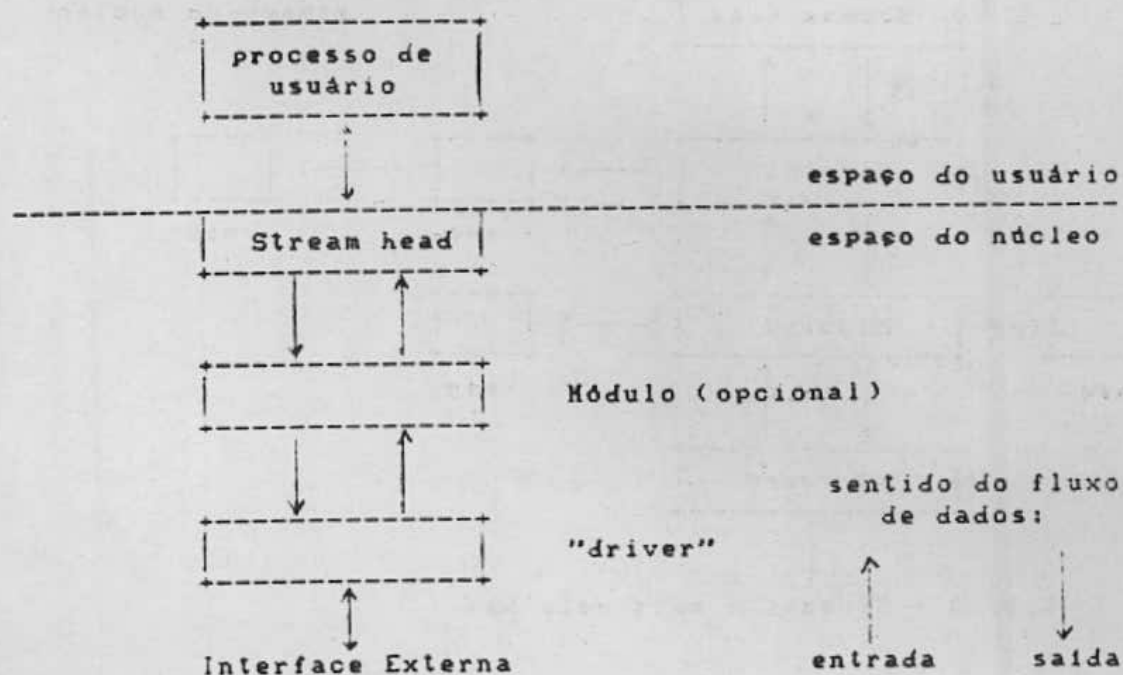


Fig. 2 - A estrutura de um Stream.

O mecanismo de STREAMS passa dados entre o "Stream head" e o "driver" na forma de mensagens, que são a única forma interna de comunicação e transferência de dados. Uma mensagem STREAMS contém dados, informações de estado ou controle ou ambos. Cada mensagem possui um indicador de tipo de mensagem que identifica o seu conteúdo.

Dados enviados a um "driver" são empacotados na forma de mensagens e fluem no sentido de saída. Mensagens que chegam ao "Stream head" vindas no sentido de entrada são processadas e os dados são copiados nos "buffers" do usuário.

O mecanismo de STREAMS é projetado baseado em filas de mensagens. Um módulo de Stream realiza as transformações intermediárias nas mensagens que fluem entre "Stream head" e "driver". Cada módulo é constituído de um par de estruturas FILA que são criadas quando o módulo é inserido num Stream. Uma FILA é responsável pelas mensagens da fila de entrada e outra pelas mensagens da fila de saída (característica de bidirecionalidade e simetria de um Stream). Neste caso, pode-se dizer que um Stream é um conjunto de filas através das quais fluem as mensagens.

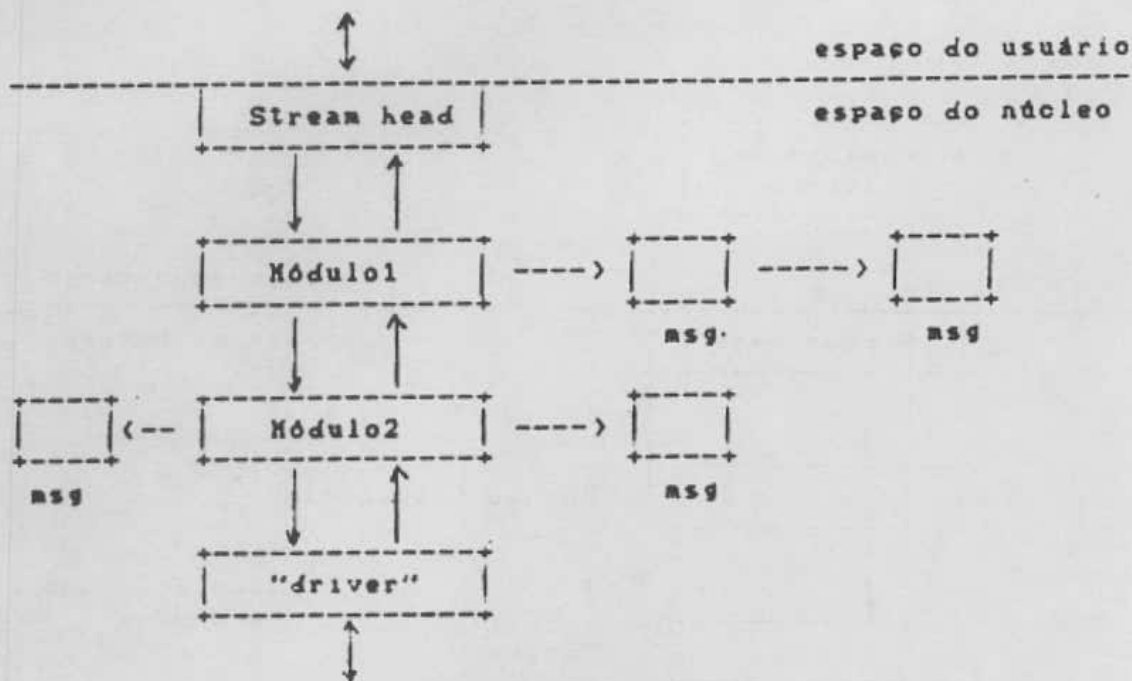


Fig. 3 - Streams - mais detalhes.

Cada estrutura FILA de um módulo pode conter ou apontar para mensagens, dados ou rotinas de processamento:



- mensagens - são associadas dinamicamente à FILA e organizadas na forma de lista ligada, conforme vão atravessando o módulo;

- dados - são dados particulares como por exemplo, informação de estado e tabelas de tradução, necessárias ao processamento das mensagens naquele módulo.

- rotinas de processamento - são rotinas de atendimento das mensagens e possuem a capacidade de enviar mensagens no dois sentidos e modificar os dados do módulo;

A principal vantagem do mecanismo de STREAMS é a sua modularidade. STREAMS pode inserir um ou mais módulos residentes no núcleo num Stream, entre o "Stream head" e o "driver", para realizar processamentos de dados intermediários. Do nível de usuário, os módulos residentes no núcleo podem ser conectados dinamicamente para implementar qualquer sequência de processamento de dados racional. Esta modularidade beneficia a implementação de serviços e protocolos de redes, refletindo nas características de organização em camadas das arquiteturas modernas de redes.

Outra vantagem de STREAMS é a sua simplicidade para a criação de módulos que apresentam serviços de comunicação de dados de forma padronizada - esta interface é especificada por um conjunto de mensagens e regras que controlam a passagem dessas mensagens através da fronteira.

A modularidade e definição de interfaces de serviços do mecanismo de STREAMS convertem-se em capacidades como:

- programas a nível de usuário podem ser independentes de protocolos e meios físicos de comunicação;

- serviços de nível mais alto podem ser criados selecionando-se e conectando-se serviços e protocolos de nível mais baixo.

- substituição de protocolos - módulos alternativos e "drivers" de dispositivos podem ser substituídos na mesma máquina se implementarem uma interface de serviço equivalente.

- portabilidade de módulos de protocolos - o mesmo módulo de protocolo pode ser usado em diferentes "drivers", em máquinas diferentes, desde que estas ofereçam interfaces compatíveis de serviços.

## 2.4 - Sistema de Arquivos Distribuído

O compartilhamento de arquivos remotos (Remote File Sharing - RFS) é uma implementação de sistema de arquivos distribuído baseado no sistema operacional UNIX.

Os objetivos principais do RFS é prover aos usuários e aplicativos um meio de acessar arquivos remotos, levando os seguintes pontos em consideração:

- acesso transparente - não há distinção para o usuário ao acessar um arquivo local ou remoto;
- preservação da semântica - todos os tipos de arquivos são acessíveis, inclusive os arquivos especiais (dispositivos) e os "pipes" nomeados. Prove suporte para travamento de arquivos e registros;
- independência de tipo de rede;
- portabilidade para diferentes implementações de "hardware".

A arquitetura do RFS é baseada num modelo cliente/servidor, usando um servidor central de nomes e um esquema de "remote mount" (como "mount" no UNIX) para estabelecer conexão entre máquinas.

Estendendo a noção de "mount/umount", RFS permite que uma sub-árvore de uma máquina servidora seja logicamente conectada à árvore de arquivos local de uma máquina cliente.

Estabelecendo a conexão lógica entre duas máquinas da rede, através dos pontos de entrada e ligação (diretório de ligação), o usuário "caminha" de uma estrutura de árvore a outra sem tomar conhecimento do desvio (a localização física do arquivo é irrelevante para o usuário).

A comunicação entre as máquinas é feita através de um protocolo de mensagens baseado na interface de chamadas de sistema do UNIX.

O mecanismo de STREAMS e a Interface de Transportes (TLI) do UNIX foram usados para separar o RFS da rede.

O mecanismo FSS (File System Switch) permite que o mesmo sistema operacional UNIX suporte implementações diferentes de sistemas de arquivos, separando as informações dependentes das informações independentes contidas no descriptor do arquivo. Define-se um descriptor tipo RFS, que contém um ponteiro de comunicação através do qual se consegue acessar o descriptor "real" de um arquivo no servidor.

O mecanismo normal de segurança e proteção de arquivos é extensível aos arquivos remotos e um mecanismo de mapeamento de identificação de usuários é fornecido.

O RFS pode ser representado como mostra a figura abaixo:

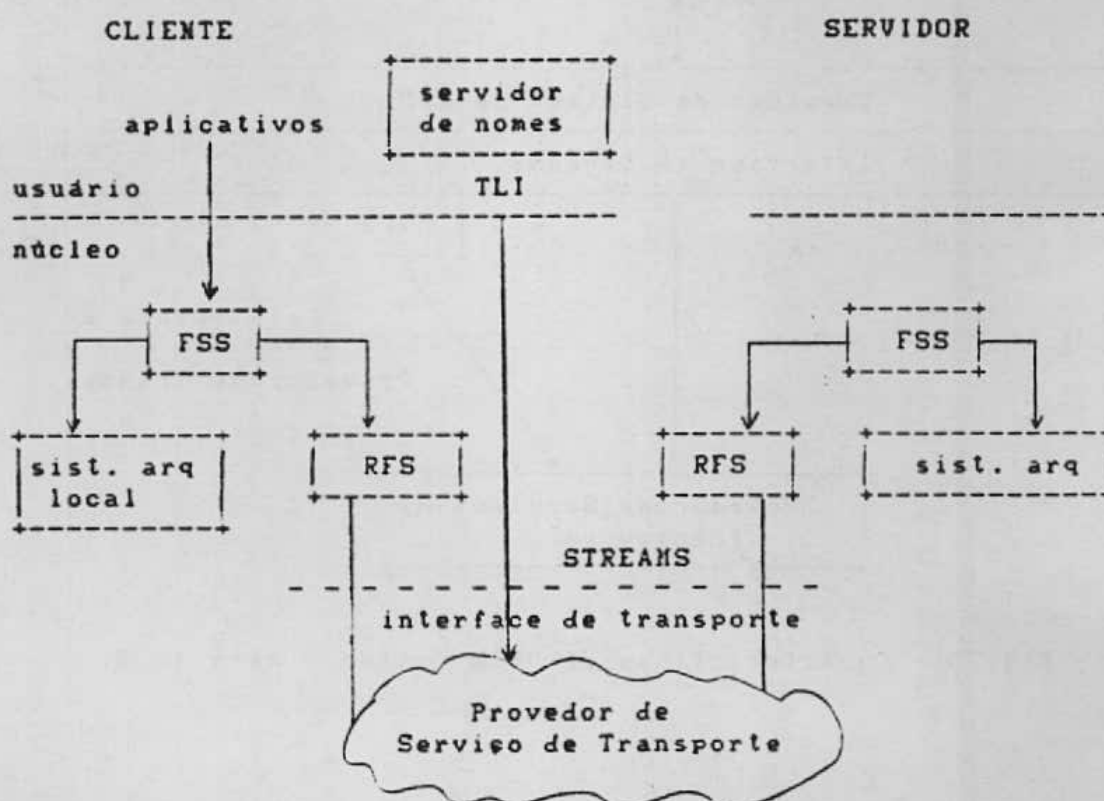


Fig. 4 - Sistema de Arquivos Distribuído do UNIX - "Remote File Sharing".

As características incorporadas à nova versão do sistema UNIX dão suporte ao desenvolvimento de protocolos e serviços de redes. A base dessas características é o mecanismo de STREAMS que, além de ser um mecanismo novo de entrada/saída do sistema UNIX, fornece meios para a construção de facilidades de rede, como por exemplo, a biblioteca de usuário fornecendo interface de serviço a nível de transporte e o sistema de arquivos distribuído para acesso transparente de arquivos remotos.

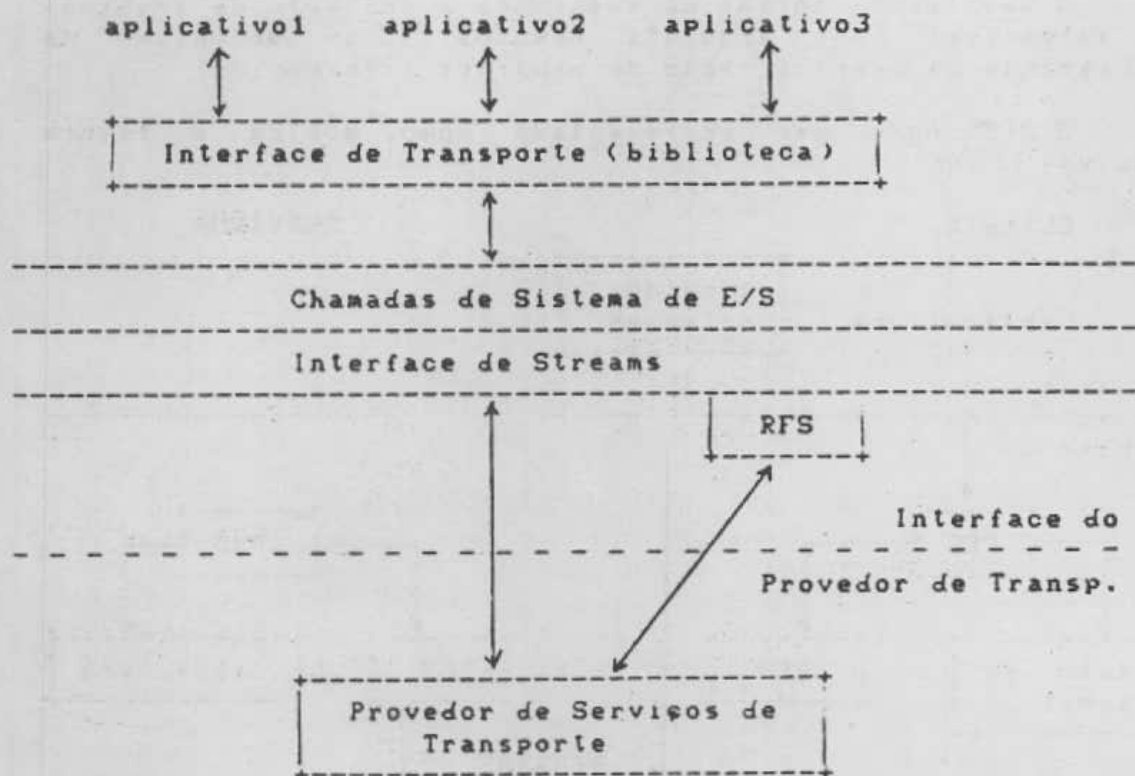


Fig. 5 - Características do UNIX System V Release 3.

### 3 - A Implementação de Rede no Mini-supercomputador MS8701

O Subsistema de Controle de Comunicações (SCC) do Minissuper, foi especificado para atender diretamente uma rede local padrão ETHERNET e uma comporta (do tipo "gateway") que permite a conexão dos usuários do sistema com uma rede pública, no padrão X.25 (RENPAK). Além do controle de terminais síncronos/assíncronos e de uma expansão de terminais assíncronos através de uma rede de teleprocessamento.

Essa rede de teleprocessamento realiza a conexão de um grande número de terminais localizados fora do gabinete principal, onde situa-se a Unidade de Processamento do Sistema (UPS). Os terminais são ligados à UPS através das Controladoras de Terminais Distribuídos (CTD's).

No gabinete principal do Minissuper localiza-se a Placa de Processamento de Comunicações (PPC), que gerencia as mensagens transmitidas/recebidas entre o computador e os diversos terminais e Placas Especializadas de Entrada/Saída (PEES), as quais estão conectadas num barramento de 32 bits de alta velocidade (padrão VSB), que se comunica com a PPC.

As CTD's permitem a ligação dos terminais assíncronos, (distribuídos ao longo da planta do edifício) à UPS através de um barramento CSMA/CD, utilizando um conjunto de protocolos baseados no Modelo OSI. Estão organizadas de forma modular, incluindo: uma Placa de Controle de Comunicação (PCC), que faz interface com a rede; e Placas de Expansão de Entrada/Saída (PES), que contém os dispositivos de comunicação serial com os terminais.

#### 3.1 - Características Gerais do "hardware"

De acordo com o diagrama a seguir, o Subsistema de Controle de Comunicações (SCC) é constituído por uma Placa de Processamento de Comunicações, no módulo principal, Placas Especializadas de Entrada/Saída, conectadas ao duto (VSB) de entrada/saída e Controladores de Terminais Distribuídos.

A PPC é constituída de um processador monolítico de 32 bits, responsável pela execução do programa de gerenciamento das portas locais da placa e das placas que constituem o subsistema, comunicando-se com os demais módulos do Minissuper através de uma memória compartilhada, e de um canal formado por dois dutos de 32 bits de dados e endereçamento, especificado a partir do padrão VME bus.

#### Características gerais da PPC:

- UCP monolítica MC68020 a 16,67 (ou 20) MHz de relógio;



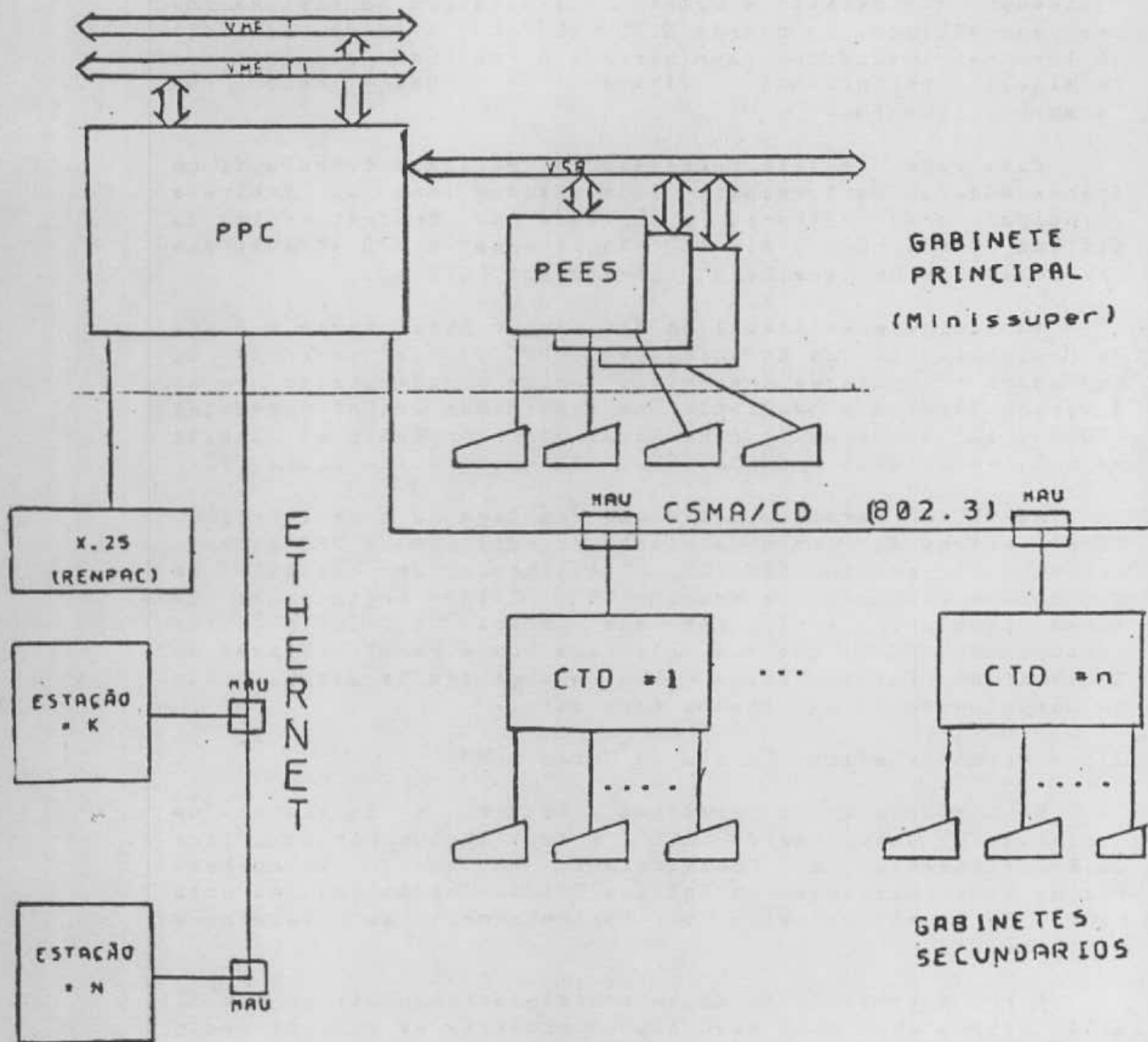


Fig 6 - Diagrama de blocos do subsistema de comunicações.

- memória fixa, EPROM de 64 Kbytes (27512) - para armazenamento do programa de "start up";
- memória de leitura/escrita, RAM estática de 128 (256) Kbytes, para armazenamento do programa de gerenciamento de E/S;
- memória compartilhada, RAM dinâmica de 4 (8) Mbytes, para comunicação com o sistema e armazenamento das filas de mensagens e dados da comunicação;
- 1 porta serial, de comunicação assíncrona, padrão RS232-C - para funções de depuração;
- interface com o canal de expansão de entrada/saída padrão VSB (Motorola);
- interface para rede de comunicação (IEEE 802.3) com controladores de terminais;
- interface mestre com dois dutos de 32 bits tipo VME bus (para transmissão de mensagens de interrupção - ou atendimento - de comandos e eventos assíncronos dos terminais e estações do subsistema).

As placas especializadas de entrada e saída (PEES) conectam-se, através do canal de E/S (duto VSB), à PPC e realizam a comunicação com os terminais pelas portas de expansão locais a essas placas.

Um microprocessador monolítico, MC68008, atua como controlador do módulo, que é constituído de:

- memória local de programa (64 Kbytes de EPROM e 64 Kbytes de RAM);
- memória de buffer para comandos/caracteres (64K estática);
- interface de E/S (serial, paralela, etc);
- interface com o canal de E/S (VSB).

A interface de E/S depende diretamente da forma de comunicação e protocolo realizado pela placa como:

- expansão de terminais de comunicação serial RS-232 / 422, em taxa máxima de 38400 bps;
- expansão de comunicação paralela de alta velocidade.

As Controladoras de Terminais Distribuídos (CTD's) são módulos da Placa de Processamento de Comunicações (PPC) controlados por uma rede local de comunicação CSMA/CD. Estes módulos são constituídos de uma placa inteligente (dotada de microprocessador) que controla um conjunto de placas de expansão de portas de E/S seriais assíncronas, as quais podem possuir um micro-controlador da interface de E/S - mas não um processador, como a placa principal da CTD.

A Placa Controladora de Comunicações (PCC), responsável por gerenciar o módulo da CTD realiza o protocolo de comunicação com a PPC, via rede local, e é dotada de:

- microprocessador MC68020 a 16,67 M Hz;
- memória de programa (64 Kbytes de EPROM e 128 Kbytes de RAM);
- memória de armazenamento dos "buffers" de caracteres dos terminais (256 Kbytes estática);
- interface com rede IEEE 802.3 de comunicações;
- interface com o canal de E/S (para comunicação com as placas que implementam as portas de E/S).

### 3.2 - Programa de Controle de E/S da PPC

O programa de controle de entrada/saída que integra a PPC é carregado, em memória RAM de acesso local do processador, no instante de ligação da máquina ("start-up") e tem por função controlar as seguintes operações:

- tratamento dos comandos para as portas de comunicação, presentes nas placas de controle especializado;
- gerenciamento de mensagens para os controladores de terminais conectados via rede local.

A comunicação de dados e mensagens para os terminais e placas especializadas, e a implementação dos diversos níveis de tratamento dos terminais virtuais será feita através da utilização do mecanismo de STREAMS.

O mecanismo de STREAMS impõe uma estrutura de módulos de tratamento das mensagens de E/S hierarquizado em níveis de procedimento de cada protocolo (o que facilita a implementação do sistema de níveis de rede).

Como mencionado anteriormente, a expansão para um maior número de terminais é feita através de controladoras (CTD's) conectadas à PPC por uma interface de rede. Esta interface define o controle das CTD's através de um protocolo de pacotes de comunicação permitindo realizar o gerenciamento desses terminais pelo sistema, utilizando operações descritas por mensagens de:

- escrita de dados em dispositivo;
- leitura de dados de dispositivo;
- escrita de configuração de porta;
- requisição de leitura de configuração de terminal;
- controle da porta e da placa (atualização do programa de controle).

O protocolo de comunicação de dados, entre as CTD's e o Mini-super, inclui os seguintes módulos:

- o "Terminal Driver": é o nível de protocolo mais alto; controla o funcionamento dos terminais e realiza as políticas de tratamento de linha;
- o Protocolo de Controle de Dispositivos (PCD): interface entre o terminal e o módulo de controle de fluxo; é responsável pelo controle da CTD, atendimento de comando de carga do código da PCC, configuração dos terminais e pedidos de leitura/escrita - as mensagens utilizadas pelo PCD são compatíveis com o protocolo X.29;
- o Protocolo de Controle de Fluxo (PCF): é o módulo de interface entre o PCD e o nível de enlace da rede; encarregado da comunicação com a rede local e controle de fluxo de mensagens num protocolo compatível com X.25;
- os níveis de enlace e método de acesso: utilizam um protocolo LLC classe 1 e uma pastilha que realiza o nível MAC (CSMA/CD).

O protocolo PCF é orientado à utilização de "janelas" de comunicação e permite a transmissão automática de um ou mais pacotes de dados, segundo a quantidade de memória disponível, entre o Minissuper e as CTD's.

A CTD devolve as seguintes mensagens de dados, configuração e eventos assíncronos dos terminais:

- resposta de leitura de dados do "buffer" do terminal;
- resposta de leitura de configuração do terminal;
- sinal de "break" recebido do terminal;
- falha em subsistema de E/S;
- time-out de comunicação de dados;
- caractere de interrupção recebido do terminal.

A CTD, portanto, deverá executar tarefas de edição de linha e orientadas a caracteres, mas as transferências entre o módulo PPC e o buffer de dados dos terminais ligados à CTD deverão ser, para diminuição do "overhead" de comunicação, empacotadas em blocos de dados.

O "terminal driver" estará encarregado da transferência de dados de/para os terminais, comunicando os terminais burros à PCC que se encarregará de executar os comandos enviados pela PPC. Na condição inicial de ligação do sistema, a PCC possui um "firmware" muito primitivo que permite:

- realizar um teste de comunicação com os terminais ligados a ela.

- a carga do código da PCC, que será executado posteriormente, através da troca de mensagens entre a PCC e sua correspondente PPC.

O protocolo de comunicação de rede da CTD procura substituir os protocolos padrões devido à objetividade e simplicidade, permitindo a orientação de tarefas a nível de caracteres. Este protocolo surge como necessidade de um mecanismo de transporte para evitar sobrecarga da CTD pelo hospedeiro, fornecer um retorno em tempo real à tecla digitada, ou para quando o usuário precisar deter o fluxo de dados por um determinado instante, etc. Estes pedidos fazem parte da definição do protocolo de controle de fluxo (PCF).

A CTD realiza todas as tarefas orientadas para caracteres. O Host (Minissuper) controla o comportamento da CTD usando o PCD, que é semelhante ao protocolo X.29. O PCD está projetado para ser simples e eficiente. A simplicidade é mostrada através de um número pequeno de tipos de mensagens com um cabeçalho de tamanho fixo.

O protocolo de controle de dispositivos está implementado num cartão multiplexador, para aproveitar os recursos do hardware, tais como capacidade dos microprocessadores e espaço em RAM. O PCD não é simétrico, pois o Host atua como mestre e a CTD como escravo.

O PCF é um protocolo orientado à conexão. Onde há um procedimento simples para estabelecer e finalizar as conexões entre a CTD e o hospedeiro. A maior utilidade deste é realizar um melhor gerenciamento de tal maneira a assegurar que a CTD não fique sobrecarregado pelo Host.

Devido ao fato da LAN restringir o número de caracteres que devem ser passados, o PCF possui um mecanismo para fragmentar e reorganizar os dados que são passados através da linha (LAN). O protocolo PCF é simétrico, não havendo uma relação mestre/escravo.

Para fornecer melhor desempenho, sem a utilização do esquema pedido/resposta, o qual é menos eficiente, é utilizado um mecanismo de janela. Que trabalha com três campos:

- o número da sequência do campo, com a identificação do pacote enviado;
- o número do reconhecimento, que indica o número de sequência que o receptor está esperando;
- o campo da janela, que indica o último número da sequência que o receptor está disposto a aceitar.



Para facilitar a manipulação, o cabeçalho do pacote do PCF tem um tamanho fixo, são utilizados diversos tipos de pacotes para estabelecer e findar conexões, e para fornecer segurança e confiança no intercâmbio de informações.

#### 4 - Conclusões

Entre os mecanismos de comunicação mais usados, podemos destacar o STREAMS, que passou a ser mais comum a partir do Release 3 do sistema operacional UNIX. Como principal característica, possui a inserção dinâmica de módulos ou rotinas configuradas para cada terminal.

Outro modelo utilizado é o Protocolo de Controle de Fluxo (PCF), que utiliza o conceito de janelas, para aproveitar ao máximo os recursos de memória disponível.

O protocolo de controle de fluxo de dados utiliza o mecanismo de troca de mensagens através da linha (LAN). Essas mensagens utilizam estruturas auxiliares para facilitar o manuseio da informação bidirecional entre o usuário e o sistema.

O sistema aproveita os mecanismos anteriormente descritos para obter maior modularidade, flexibilidade e velocidade na transferência de informações entre os diversos dispositivos que fazem parte da rede.

A rede de teleprocessamento baseada em Controladoras de Terminais Distribuídos (CTD's) procura introduzir facilidades de protocolos com características de segurança, simplicidade de comunicação, fragmentação e reorganização de mensagens em pacotes; pretende também atender à necessidade de grande número de terminais, visando diminuir os custos de instalação envolvidos.

A rede local, entre o Minissuper e as estações gráficas, irá permitir a implementação de um Sistema de Arquivos Distribuído, de modo que os usuários possam usufruir dos recursos da rede de forma eficiente e econômica.

Os sistemas operacionais desenvolvidos no LSI, por serem compatíveis à versão anterior ao UNIX System V Release 3, não possuem nenhuma das facilidades incorporadas. O projeto de redes do LSI pretende, partindo-se dos trabalhos desenvolvidos no sistema operacional e mantendo a compatibilidade com o sistema UNIX, fornecer ferramentas para a construção de uma rede de computadores interligando as máquinas UNIX desenvolvidas no LSI.

## 5 - Bibliografia

AT&T, "UNIX System V Network Programmer's Guide", Prentice-Hall, Inc, 1987.

AT&T, "UNIX System V STREAMS Primer", Prentice-Hall, Inc, 1987.

AT&T, "UNIX System V Streams Programmer's Guide", Prentice-Hall, Inc, 1987.

Brown, L. M. "Networking Architecture & Protocol", em "UNIX System Software Readings", Prentice-Hall, Inc, 1987, pag. 81-107.

Buchanan et. al. "A Distributed Terminal Controller for HP Precision Architecture Computers running the MPE XL Operating System" - HP Journal / March 1987

Chappell, D. "Guide to transport-layer interfaces for Unix users"- Data Communications/ July 1987

Emrich, J. "Remote File System, Streams, and Transport Level interface", em "UNIX Papers" Edited by The Waite Group, Howard W. Sams & Company, 1987, pag. 260-305.

HP Journal - Edição especial - Outubro 1986

McGrath, G. J. "Streams Technology", em "UNIX System Software Readings", Prentice-Hall, Inc, 1987, pag. 49-79.

Nidorikawa, E. T. et. al., "LSI-SO.01 - Um Sistema Operacional Multiprocessador" Anais do II SBAC-PP, Aguas de Lindóia, São Paulo (Setembro 1988).

Kokhoff, M. "Five-chip token-passing set operates LANS at 100 Mbits/s" - Electronic Design / September 17, 1987

Olander, D. J. et. al., "A Framework for Networking in System V" USENIX Conference Proceedings, Atlanta, Georgia (June 1986).

Rifkin, A. P. et. al., "RFS Architectural Overview" USENIX Conference Proceedings, Atlanta, Georgia (June 1986).

Ritchie, D. M. "A Stream Input-Output System", AT&T Bell Laboratories Technical Journal 63(8) (October 1984).

Sabsevitz, A. L. "Distributed UNIX System - Remote File Sharing", em "UNIX System Software Readings", Prentice-Hall, Inc, 1987, pag. 109-157.

Smith, B. R. "Managing BSD Unix Networks" - Unix World /  
September 1988

Stallings, W. "Local Networks" - Computing Surveys / vol.  
16 March 1984