

Redes de Computadores

Teoria e Prática

Conteúdo

Parte1. Introdução aos Conceitos Fundamentais em Redes de Computadores

1.1 Conceitos Básicos

- Estrutura de Redes
- Arquitetura de Redes
- Serviços
- Meios de Transmissão
- Transmissão Digital

1.2 A Subcamada de Acesso ao Meio

- Alocação de Canais
- Protocolos de Camadas 1 e 2
- Padrão IEEE 802 para Redes
 - * ethernet padrão
 - * ethernet a 100 Mbps
 - * token ring
- Redes de Fibra Ótica
- ATM
- FDDI

Parte 2. Interconexão de Redes e Projeto de Instalação

- 2.0 Hubs e Switches
- 2.1 Repetidores
- 2.2 Pontes e Roteadores

Parte 3. Introdução aos Protocolos e Serviços Internet

- 3.0 A tecnologia Internet
- 3.1 Os protocolos Principais: IP, ARP, TCP e UDP
- 3.2 Serviços: Telnet, FTP, WWW, Gopher, XArchie, Netfinder

Parte 4. Instalação de Rede e Serviços de Rede

- 4.1 Configuração e Instalação de uma Subnet de Comunicação**
 - 4.1.1 Cuidados Iniciais
 - 4.1.2 Configuração do Hardware
 - 4.1.3 Configuração do Software (Pocket Driver e KA9q)
- 4.2 Instalação, Configuração e Utilização de uma API TCP/IP em Windows**
 - 4.2.1 As camadas
 - 4.2.2 Preparação de Arquivos para os Clientes Novel e TCP/IP
 - 4.2.3 Instalação de um Cliente Novel
 - 4.2.4 Upgrade para um Cliente TCP/IP

Parte 1. Introdução aos Conceitos Fundamentais em Redes de Computadores

Livro Referência: Computer Networks, 2nd. edition, Andrew Tanenbaum, Prentice-Hall 1989. Esta parte da apostila apresenta um resumo dos pontos principais abordados nos capítulos I, II, III. Este livro é uma fonte de consulta muito boa, pois coloca a disciplina sobre conceitos bem definidos. A PC-Magazine tem dois livros lançados (em Português) que cobrem a mesma matéria, porém de maneira mais técnica. São: Guia de Conectividade e Guia para a Interligação de Redes Locais, da editora Campos.

1.1 Conceitos Básicos

O ponto chave da tecnologia dominante neste século tem sido a aquisição, o processamento e a distribuição da informação.

- Instalação de redes telefônicas de alcance mundial.
- Invenção do rádio e televisão.
- Nascimento e o contínuo crescimento da indústria de computadores.
- Lançamento de satélites de comunicação.

Nos dias atuais as áreas descritas acima estão convergindo rapidamente e as diferenças entre coleta, transporte, armazenamento e processamento de informação desaparecem rapidamente.

Durante as duas primeiras décadas de sua existência, os sistemas computacionais eram altamente localizados (normalmente dentro de uma grande sala) - o Centro de Computação.

Este modelo apresenta dois problemas:

1. O conceito de um único e grande computador fazendo todo o trabalho
2. A ideia dos usuários trazerem o trabalho ao computador, ao invés de levar o computador ao usuário

Este modelo arcaico está sendo rapidamente trocado por sistemas em que um grande número de computadores separados, mas interconectados, fazem a tarefa. É o que chamamos de **Rede de Computadores**.

Redes de Computadores: uma coleção de computadores autônomos interconectados.

A Utilização das Redes de Computadores

Objetivos:

1. Fazer todos os programas, dados e outros recursos disponíveis a todos, sem se considerar a localização física do recurso e do usuário.

2. Barateamento de processamento.

Até 1970, computadores eram muito caros se comparados aos custos de comunicação. Atualmente, a situação se inverte: é mais barato colocar vários computadores para análise local de dados com transmissão eventual destes dados.

3. Fornecer um meio de comunicação eficiente entre pessoas trabalhando distantes umas das outras.

4. Adicionalmente pequenos computadores tem uma taxa preço/performance muito melhor que os *mainframes*. Os *mainframes* são mais ou menos 10 vezes mais rápidos que os micro-computadores, mas são mais ou menos 1000 vezes mais caros.

Isto favorece a criação de redes locais de computadores (uma coleção de computadores colocados perto uns dos outros), ao invés de um *mainframe* num CPD.

Distância	Localização	Exemplo
0.1 m	Placa de Circuito	Data Flow
1 m	Sistema	Multiprocessador
10 m	Sala	Redes Locais (LAN)
100 m	Prédio	
1 Km	Campus	
10 Km	Cidade	Redes de Longa Distância (WAN)
100 Km	País	
1000 Km	Continente	
10.000 Km	Planeta	Interconexão de WANs

Estrutura de Redes

Em qualquer rede existe uma coleção de máquinas que podem rodar programas aplicativos. Chamaremos estas máquinas de *hosts*.

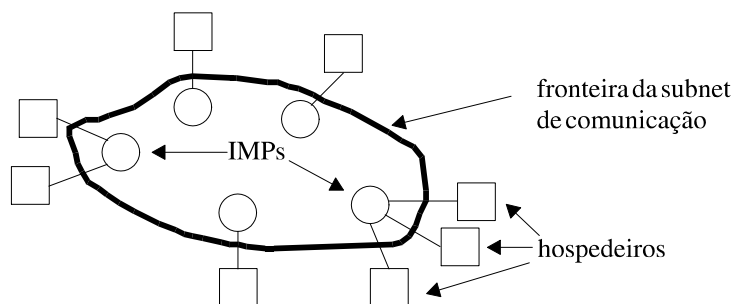
Os *hosts* são conectados pela *subnet*.

A tarefa da *subnet* é carregar mensagens de *hosts* para *hosts*.

Separando-se os aspectos de uma comunicação (a *subnet*) dos aspectos aplicativos (os *hosts*), o projeto da rede fica simplificado.

Subnet:

- Elementos chaveados: computadores especializados , IMP (*Interface Message Processor*).
- Linhas de transmissão: circuitos ou canais.



Existem basicamente dois tipos genéricos de projetos para a *subnet* de comunicação:

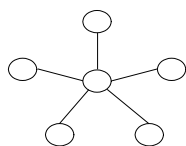
1. Canais ponto-a-ponto
2. Canais de difusão (*broadcast*)

Numa *subnet* ponto-a-ponto, a rede contém inúmeros cabos (ou linhas telefônicas privativas), cada um conectando um par de IMPs.

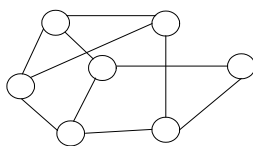
Se dois IMPs que não compartilham um cabo querem se comunicar, eles devem fazê-lo indiretamente via outros IMPs.

Subnets usando este princípio são chamadas *store-and-forward*.

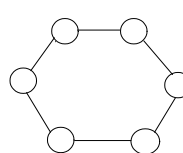
Várias topologias podem ser utilizadas para este tipo de *subnet*.



estrela



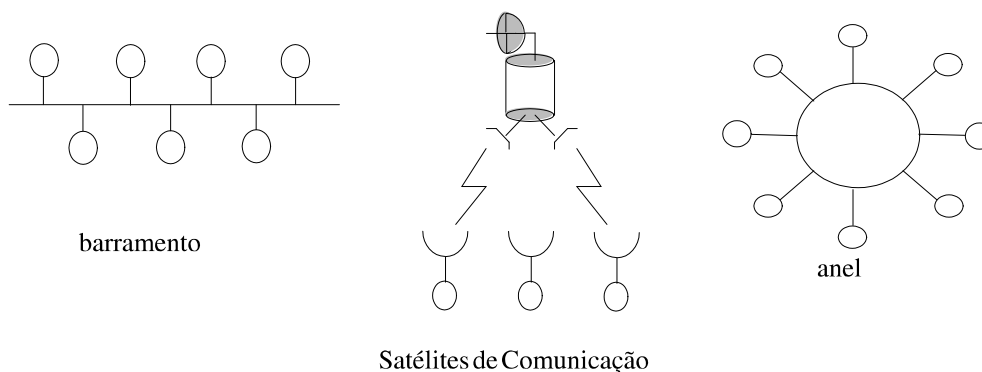
irregular



anel

No caso das *subnets* tipo *broadcast*, existe um único canal de comunicação que é compartilhado por todos IMPs. Uma mensagem enviada por um IMP é recebida por todos os outros IMPs. Alguma coisa na mensagem deve especificar o destinatário.

Podem ser baseadas em:



Redes *Broadcast* podem ser divididas em:

- Estáticas: Divisão do tempo em intervalos discretos (*slots*) permitindo cada máquina transmitir apenas durante seu *slot*.
- Dinâmicas: Alocação do canal por demanda.
 - Centralizada: sistema de arbitragem único.
 - Descentralizada: cada máquina decide por si mesma.

Arquitetura de Redes

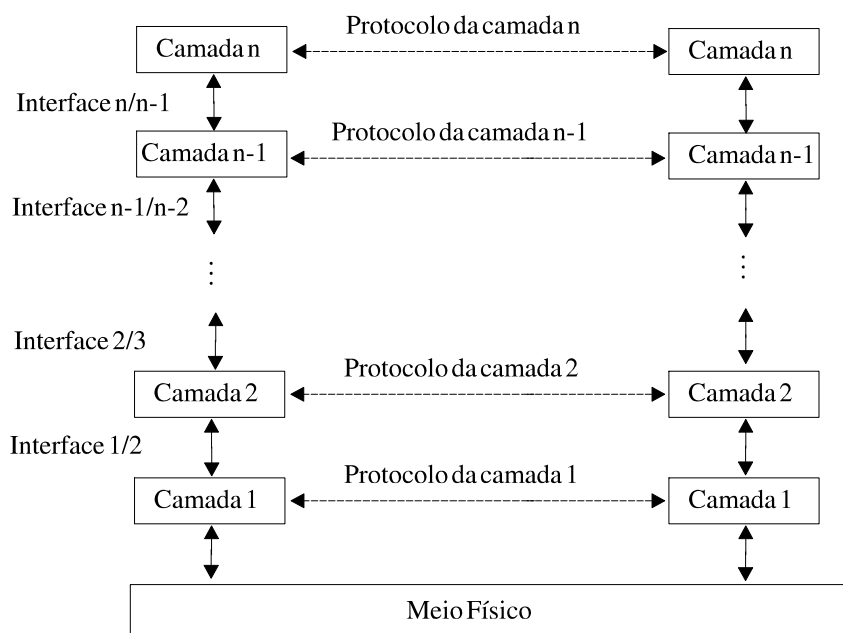
Para reduzir a complexidade, as redes são organizadas como uma série de camadas ou níveis formando uma pilha, onde:

- Número de camadas;
- Nome de cada camada;
- Conteúdo de cada camada;
- Função de cada camada

diferem de rede para rede.

A camada **n** numa máquina "conversa" com a camada **n** em outra máquina. As regras utilizadas nesta conversação são coletivamente chamadas de protocolo de comunicação da camada **n**.

As entidades que executam camadas correspondentes em máquinas diferentes são chamadas **processos pares** (*peers*). Os *peers* se comunicam usando um protocolo.

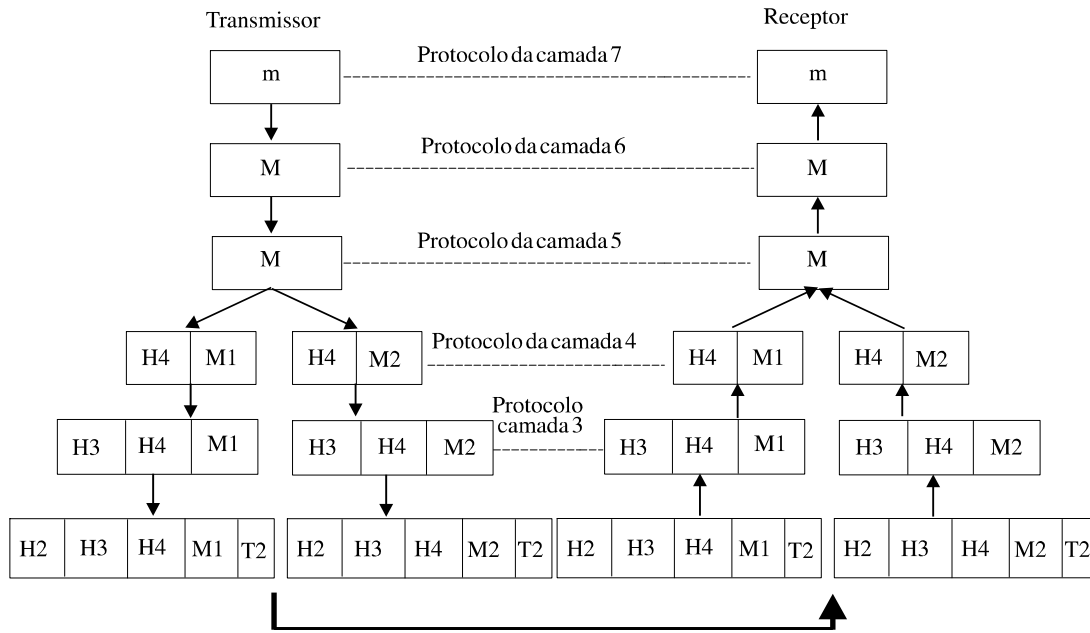


Entre cada par de camadas adjacentes existe uma interface. A interface define quais operações primitivas são oferecidas pela camada inferior para a camada superior.

Deve existir uma clara interface entre as camadas.

O conjunto de camadas e protocolos é chamado **Arquitetura de Rede**.

Fragmentação e *headers*



Os *headers* contêm informações sobre a unidade a ser transmitida. Nenhum *header* para camadas abaixo de *n* é passado para a camada *n*.

Cada *peer* imagina que se comunica horizontalmente.

Parâmetros no Projeto das Camadas

1. Mecanismos para estabelecer conexão e desconexão.
2. Regras para transferência de dados:
 - simplex - apenas numa direção.
 - half-duplex - em ambas as direções, mas não simultaneamente.
 - full-duplex - em ambas as direções simultaneamente.
3. Controle de erro (detecção e correção)
4. Sistema de ordenação de mensagens.
5. Controle de Fluxo.
6. Problema da multiplexação de canais.
7. Problema de roteamento.

O Modelo de Referência ISO/OSI

Open Systems Interconnection da International Standards Organization

O modelo ISO/OSI não é uma arquitetura de rede porque ele não especifica exatamente os serviços e protocolos a serem usados em cada camada.

A Camada Física

Esta camada está relacionada com a transmissão simples de bits sobre um canal de comunicação.

Questões típicas nesta camada:

- voltagem para bit "1"
- voltagem para bit "0"
- tempo de duração de um pulso
- o modelo de transmissão (simplex, half-duplex, full-duplex)
- como a conexão é estabelecida e cortada
- pinagem dos conectores

A Camada Link de Dados

A tarefa desta camada é tornar um sistema de transmissão cru e transformá-lo numa linha que se mostra livre de erros de transmissão à camada *network*.

Organiza a entrada em *data frames* (algumas centenas de bits), transmite os frames sequencialmente e procura frames de aviso de recebimento para enviar de volta ao transmissor.

Coloca sinalizadores de início e fim de dados.

Resolve problemas de danificação, perda e duplicação de frames.

Deve tratar do problema de conexão de máquinas de diferentes velocidades.

A Camada Network

Esta camada controla a operação da *subnet*. Sua tarefa principal é:
Como os pacotes de informação são roteados da fonte para o destino.

Rotas podem ser:

- estáticas: são definidas por hardware e são raramente modificadas.
- podem ser definidas no início de uma sessão.
- podem ser altamente dinâmicas, modificando-se a cada transmissão.

Outras tarefas:

- Controle de congestionamento e tráfego.
- Estatística de uso por usuário.
- Quando um pacote viaja de uma rede para outra, muitos problemas de compatibilidade podem aparecer (endereçamento, tamanho, etc.). A camada *network* deve resolver estas incompatibilidades.
- Em redes tipo *broadcast*, o problema de roteamento é simples, de modo que a camada *network* é muito pequena, ou mesmo inexistente.

A Camada de Transporte

A função desta camada é pegar os dados da camada de sessão, quebrá-los em partes menores, se necessário, passá-los para a camada *network* e garantir que as partes cheguem em ordem do outro lado.

Esta camada isola as camadas superiores das mudanças inevitáveis no hardware.

Cria uma conexão distinta na *network* para cada conexão requisitada pela camada de sessão.

No caso de uma requisição para conexão de grande desempenho, a camada de transporte pode criar múltiplas conectivas na *network*. A multiplexação também é feita aqui.

Pode fazer difusão de mensagens para múltiplos destinatários.

A camada de transporte é a primeira camada fonte-destino, ou seja, um programa na máquina fonte conversa diretamente com um programa na máquina destino. Nas camadas inferiores, os protocolos são entre cada máquina e seu vizinho imediato.

Muitos *hosts* permitem multiprogramação, o que implica que múltiplas conexões podem estar entrando e saindo de cada *host*. O *header* do transporte diz qual mensagem pertence a qual conexão.

A Camada de Sessão

A camada de sessão permite usuários em máquinas diferentes estabelecerem sessões (por exemplo, login, transferência de arquivos) entre elas.

Um serviço oferecido por esta camada é o controle de diálogo.

Para alguns protocolos, é essencial que ambos os lados não tentem a mesma operação ao mesmo tempo. Um sistema de *tokens* pode ser gerenciado pela camada de sessão.

Numa transferência, o problema de sincronização deve ser elaborado.

A Camada de Apresentação

Trata da sintaxe e semântica da informação transmitida.

Por exemplo, trata da codificação dos dados numa forma padrão. Faz também compressão de dados e criptografia para garantir privacidade.

A Camada de Aplicação

Contém uma variedade de protocolos que são comumente necessários:

- tipos de terminais; tipos de convenções de nomes em transferência de arquivos; correio eletrônico, etc..

Serviços

Terminologia:

Entidade: elementos ativos em cada camada

- um processo
- um chip de I/O

Entidades pares: entidades da mesma camada em máquinas diferentes.

Entidade da camada N (Fornecedora de Serviços)	<u>Implementa serviços</u>	Usados pela camada N+1 (Usuária de Serviços)
---	-----------------------------------	---

Serviços estão disponíveis nas **SAPs** e cada **SAP** tem um endereço que a identifica.

Serviços Orientados À Conexão E Serviços Connectionless

Serviço orientado à conexão: como no sistema telefónico (disca, convers, desliga).

Serviço desconectado: como no sistema postal. Toda carta carrega a identificação do destinatário.

Qualidade do Serviço:

- Confiável: nunca perde dado.
- Não confiável: para serviços onde a perda de poucas informações não causa problemas.

Podemos ter serviço:

- conectado+confiável (transferência de arquivos)
- conectado+não confiável (voz)
- desconectado+confiável (carta com aviso de recebimento)
- desconectado+não confiável (chamada **datagrama** em analogia com telegrama)

Primitivas Dos Serviços

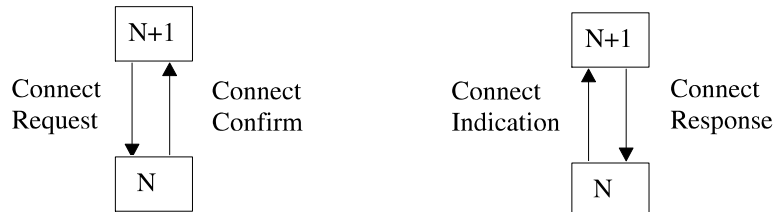
Um serviço é formalmente especificado por uma série de primitivas (operações).
Primitivas no modelo OSI:

REQUEST-uma entidade quer o serviço para executar alguma tarefa.

INDICATION-uma entidade deve ser informada sobre o evento.

RESPONSE-uma entidade que responde a um evento.

CONFIRM-uma entidade deve ser informada sobre um pedido.



Primitivas podem ter parâmetros:

Connect.Request:

- Especificação da máquina
- Tipo de serviço
- Tamanho máximo da mensagem

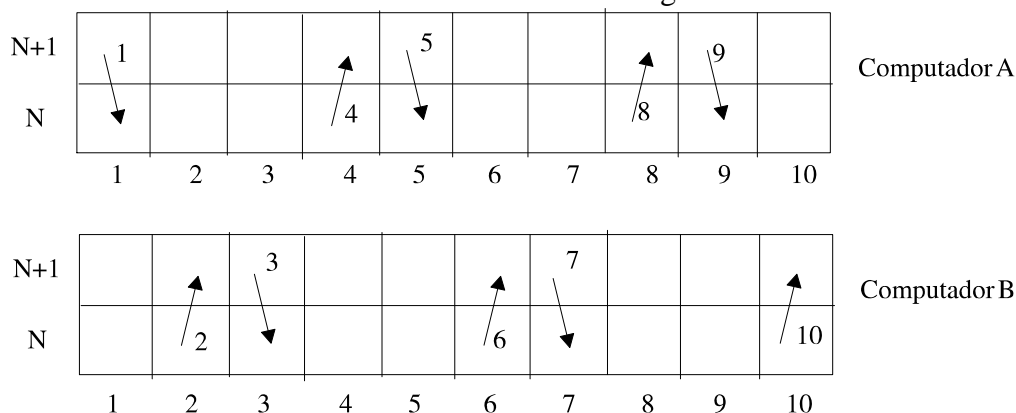
Connect.Indication:

- Identidade do que chama
- Tipo de serviço
- Tamanho máximo da mensagem

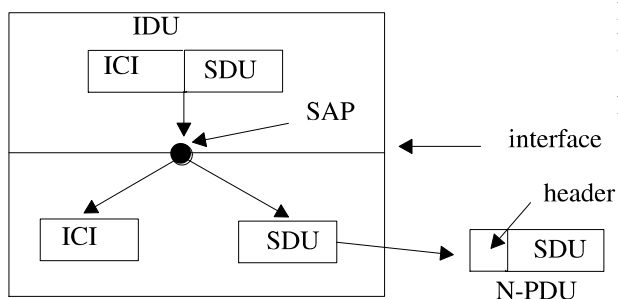
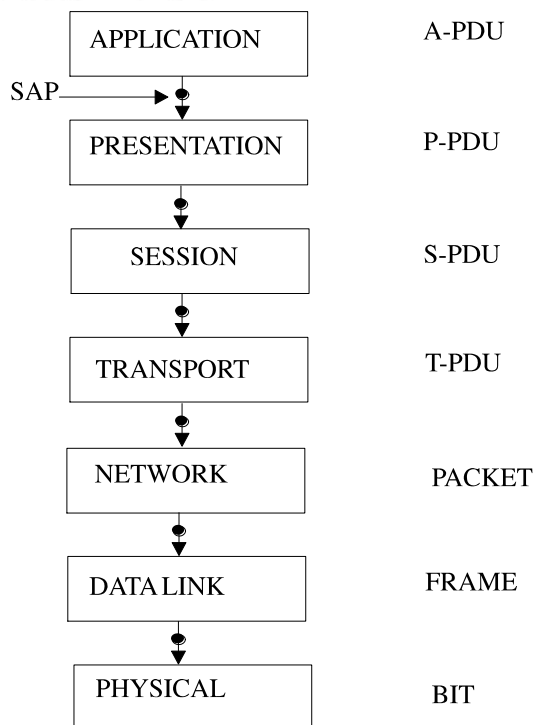
Exemplo:

Como seria uma ligação telefônica para convidar alguém para jantar, no OSI.

- CONNECT.REQUEST-Você disca um número
- CONNECT.INDICATION-O telefone toca
- CONNECT.RESPONSE-Alguém atende
- CONNECT.CONFIRM-Você percebe que o telefone parou de tocar
- DATA.REQUEST-Você faz o convite
- DATA.INDICATION-Ela ouve o convite
- DATA.REQUEST-Ela diz que gostou muito
- DATA.INDICATION-Voce ouve ela aceitando
- DISCONNECT.REQUEST-Voce desliga
- DISCONNECT.INDICATION-Ela ouve e desliga



Terminologia no modelo OSI



IDU: Interface Data Unit
ICI: Interface Control Information
SDU: Service Data Unit
PDU: Protocol Data Unit

SAP: Service Access Point

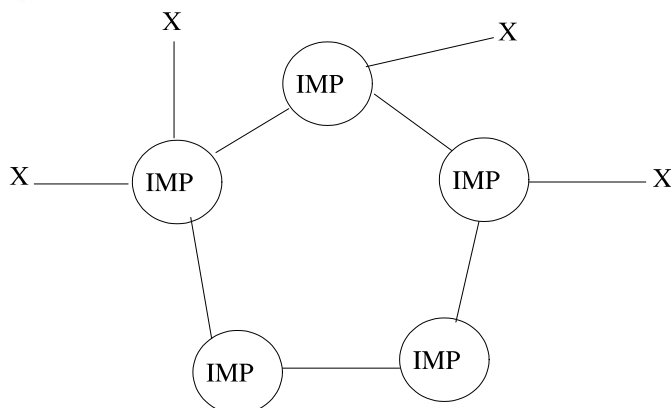
Alguns Exemplos de Redes

- Redes Públicas de Comunicação**

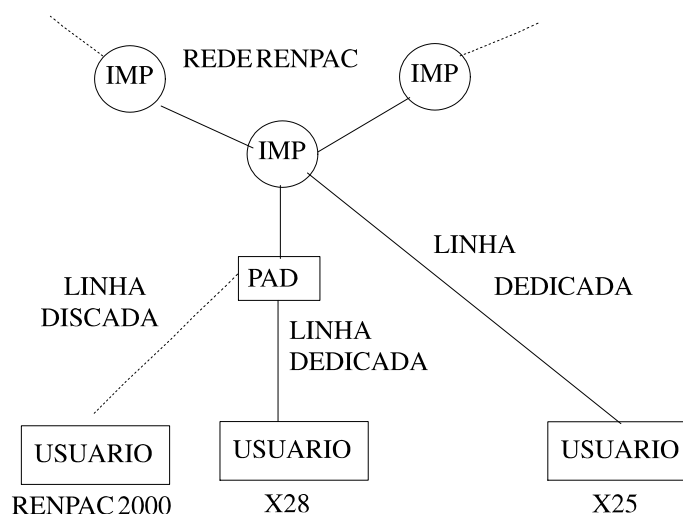
A *subnet* de comunicação pertence ao operador, por exemplo a Embratel.

Hosts e terminais pertencem aos clientes.

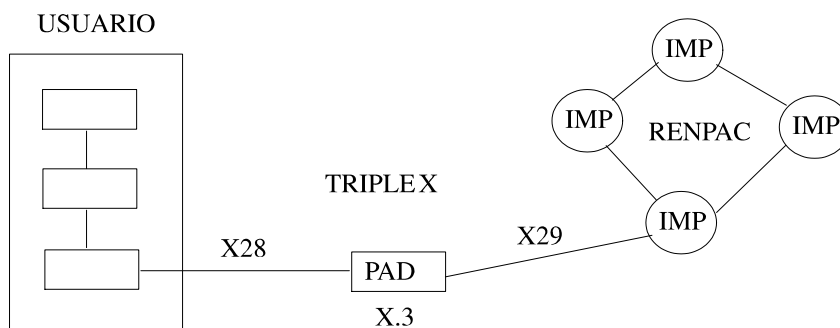
RENPA



3 MODOS DE ACESSO 'A RENPAC



Para as três camadas mais baixas (*subnet*): X25



Protocolos ao nível de aplicação:

FTAM (File Transfer, Access and Management): fornece meios para transferir, acessar e manipular arquivos remotos.

MOTIS (Message-Oriented Text Interchange System): utilizado para correio eletrônico. Similar ao X400.

VTP (Virtual Terminal Protocol): uma definição de terminal independente que capacita programas acessarem terminais remotos.

JTM (Job Transfer and Manipulation): é utilizado para submissão de processos para serem processados remotamente em *batch*.

- **Internet**

Internet não segue o modelo OSI. É anterior a ele.

IMP-IMP: mistura de protocolos das camadas 2 e 3. A camada 3 tem um esquema de roteamento bem elaborado.

Na camada de rede tem o IP e na camada de transporte o TCP: TCP/IP.

Não existem as camadas de sessão e apresentação.

Aplicativos:

- FTP-(File Transfer Protocol)
- SMTP-(Simple Mail Transfer Protocol)
- TELNET-(Login remoto)
- WWW

- **MAP e TOP**

Ambos seguem o modelo OSI.

MAP: Surgiu na GM que estava interessada em automatizar o processo de fabricação (robos interconectados). Assim sendo, a principal preocupação era a garantia de uma figura de pior caso no tempo de transmissão. TOKEN BUS foi definido a nível de *subnet*.

TOP: Surgiu na BOEING, que estava interessada na automação de escritórios. Eles não tem restrições de resposta em tempo real e usava ethernet como *subnet*. Podem também utilizar *token ring*.

Apesar de diferirem a nível de *subnet* de comunicação, são compatíveis a nível médio e alto.

- **USENET**

Baseado em princípio em UUCP (Unix-to-Unix Copy), programa desenvolvido no Unix.

Simples, se baseia em conexões sobre linhas telefônicas (normalmente tarde da noite). 10.000 máquinas conectadas.

Não existe controle central.

Único serviço: correio eletrónico.

A rede USENET é uma rede irmã do UUCP. Oferece *Network News*. A maioria das máquinas pertencentes à UUCP também pertencem à USENET.

Usuários da USENET podem se inscrever aos grupos que lhes interessam e portar mensagens que são normalmente transferidas por UUCP.

- **CSNET (hoje NFSNET)**

Metanetwork (usa as facilidades de outras redes e adicionam uma camada no topo).

Criada para interconectar Departamentos de Computação.

As redes básicas são: ARPANET, X25, PHONENET, CYPRESS.

- **SNA (Systems Network Architecture) IBM 1974-1985**

OSI copia vários conceitos da SNA: as camadas, número de camadas e funções aproximadas.

A intenção é fornecer aos clientes a possibilidade de construir na própria rede: *hosts e subnet*.

Como um dos principais objetivos era o de manter compatíveis a maioria dos protocolos lançados previamente pela companhia, o protocolo é razoavelmente complicado.

1.2 A Camada Física

Base Teórica Para Comunicação de Dados

Uma informação pode ser transmitida por fios elétricos pela variação de uma propriedade física qualquer como a voltagem ou a corrente.

Sinais podem ser representados como uma função "f (t)", onde o valor da voltagem ou corrente varia com o tempo. Assim eles podem ser analisados matematicamente.

Análise de Fourier (1904)

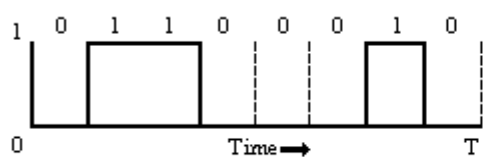
Qualquer função g(t) periódica com o período T pode ser escrita como uma soma de senos e cossenos.

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi nft) + \sum_{n=1}^{\infty} b_n \cos(2\pi nft), \text{ onde}$$

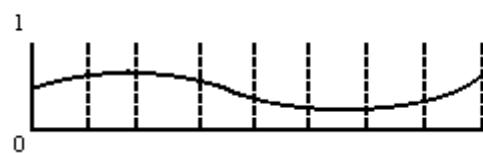
f = 1/T (frequência fundamental)

a_n, b_n são as amplitudes dos senos e cossenos da n-ésima harmônica.

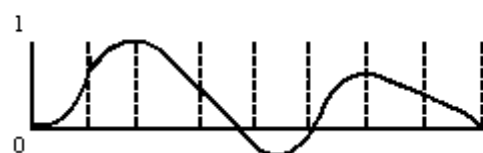
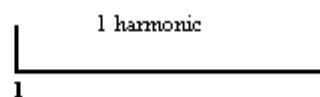
Para qualquer g(t), a, b e c podem ser calculados.



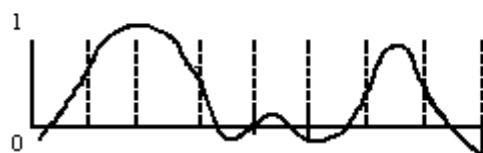
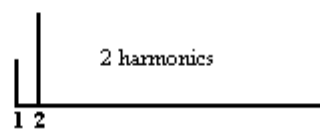
(a)



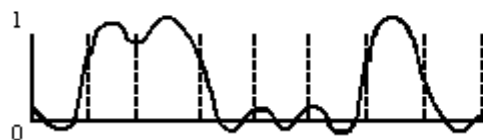
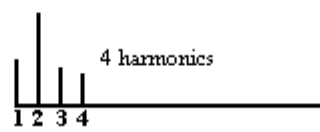
(b)



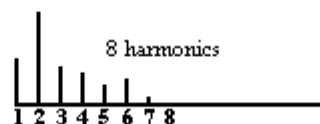
(c)



(d)



(e)



Largura de Banda

Nenhum sistema transmite sinais sem perdas de energia no processo. Adicionalmente, as perdas ocorrem de maneira diferente para diferentes harmônicas, o que insere distorção.

Normalmente, as frequências são transmitidas sem alterações até uma determinada frequência f_c . As frequências acima de f_c são fortemente atenuadas.

O limite f_c , muitas vezes é devido à propriedades físicas do meio. Em outros casos, é intencionalmente colocado na linha.

No caso de linhas telefônicas comuns, $f_c = 3 \text{ KHz}$.

"BAUD" é o número de vezes que um sinal pode mudar por segundo numa linha de comunicação.

Velocidade Máxima de Transmissão de um Canal:

- Para linhas sem ruído : Teorema de Nyquist.

$$\text{velocidade máxima} = 2H \log_2 V \text{ bits/seg}$$

onde H é a largura máxima de banda e V é o número de níveis discretos.

Para linha telefonica com $f_c = 3 \text{ KHz}$, velocidade máxima = 6 Kbps.

- Para linhas com ruído : Teorema de Shannon.

$$\text{velocidade máxima} = H \log_2 (1 + s/n)$$

Relação Sinal-Ruído : Potência do Sinal (s)/Potência do Ruído (n)

Decibel (dB) : $10 \log_{10} (S/N)$

Numa linha telefonica com $f_c = 3 \text{ KHz}$ e 30 dB, temos

max rate = 30 Kbps

independente do número discreto de níveis.

Meios de Transmissão

- Meios Magnéticos:

Gravação em fita ou disco magnético e transporte físico.

- Oferece altas taxas de transmissão
- Baixo custo po bit transportado

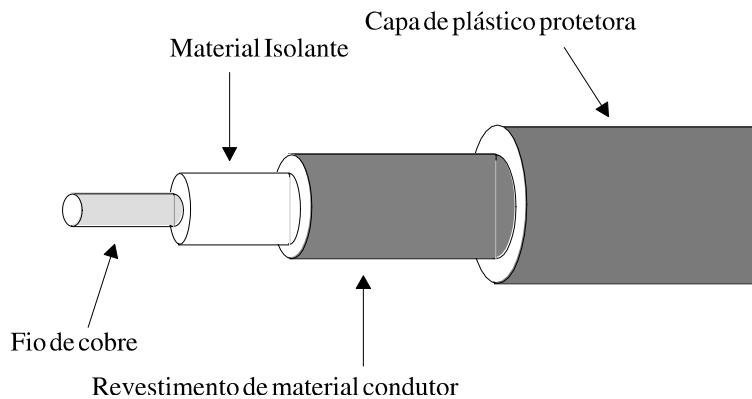
- **Par Trançado:**

Muitas aplicações precisam ter uma conexão física.

O par trançado consiste num par de fios que é trançado para evitar interferência elétrica de outros fios em volta.

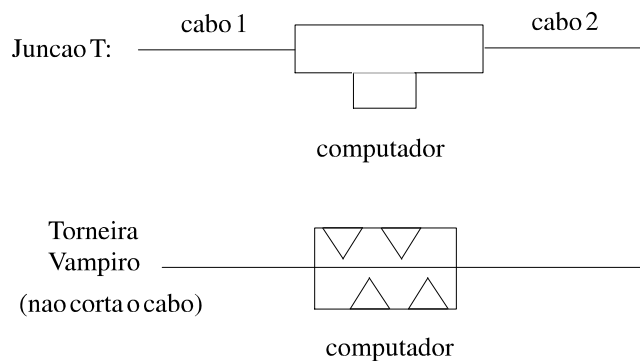
Pode ter comprimentos de ordem de quilômetros, sem amplificação. A taxa de transmissão depende da espessura do cabo e do comprimento. Recentemente foi adotado como um dos padrões de meio de transmissão para redes ethernet.

- **Cabo Coaxial:**



Baseband - 50 ohms - Transmissão digital
Broadband - 75 ohms - Transmissão Analógica.

- **Conexão computador - cabo coaxial**



"Manchester Encoding": o período de um bit é dividido em 2 intervalos iguais:

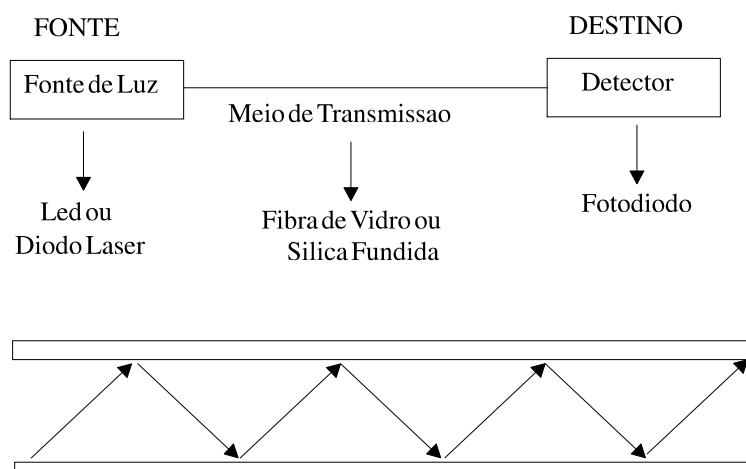
Bit 1 : HIGH	1o Intervalo	Bit 0 : LOW	1o Intervalo
LOW	2o Intervalo	HIGH	2o Intervalo

• Fibra Óptica

Dados são transmitidos por pulsos de luz., sendo que um pulso de luz corresponde ao bit "1" e a ausência de luz ao bit "0".

Potencial - 10^8 MHz

Componentes de um sistemas de transmissão :

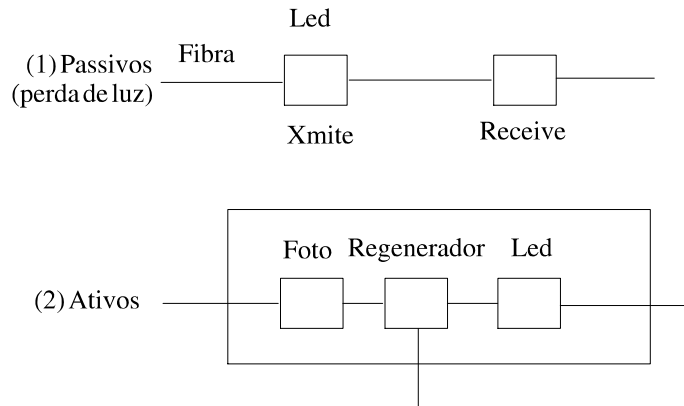


"Multimode Fiber " : os raios incidentes pulam de uma borda para outra da fibra.

"Singlemode Fiber": O diâmetro da fibra é reduzido ao comprimento de onda de luz. A luz se propaga em linha com o condutor. (Mais caro, mais eficiente, mais longo).

Fibras ópticas têm substituído os cabos em linhas telefônicas. Em LANs, o problema maior é a perda de luz quando se faz um "tap". Por isto, estas implementações utilizam topologias em estrela ou em anel (implementado como uma estrela), ao invés de duto.

Conexões:



- **Transmissão via Atmosfera.**

Dados são transmitidos via ar : Infravermelho, laser, microondas, rádio.

- **Satélites de Comunicação :**

Podem ser considerados como repetidores de microondas no céu.

Contendo um ou mais "Transponders", cada um ouvindo uma porção de espectro, amplificando e retransmitindo em outra frequência (para evitar interferencia).

O feixe descendo pode ser "Broad" (cobrindo uma faixa longa de terra) ou "narrow" (cobrindo uma área de centenas de Kms de diâmetro).

Distância mínima entre satélites : 4 graus.

Transmissão de Dados

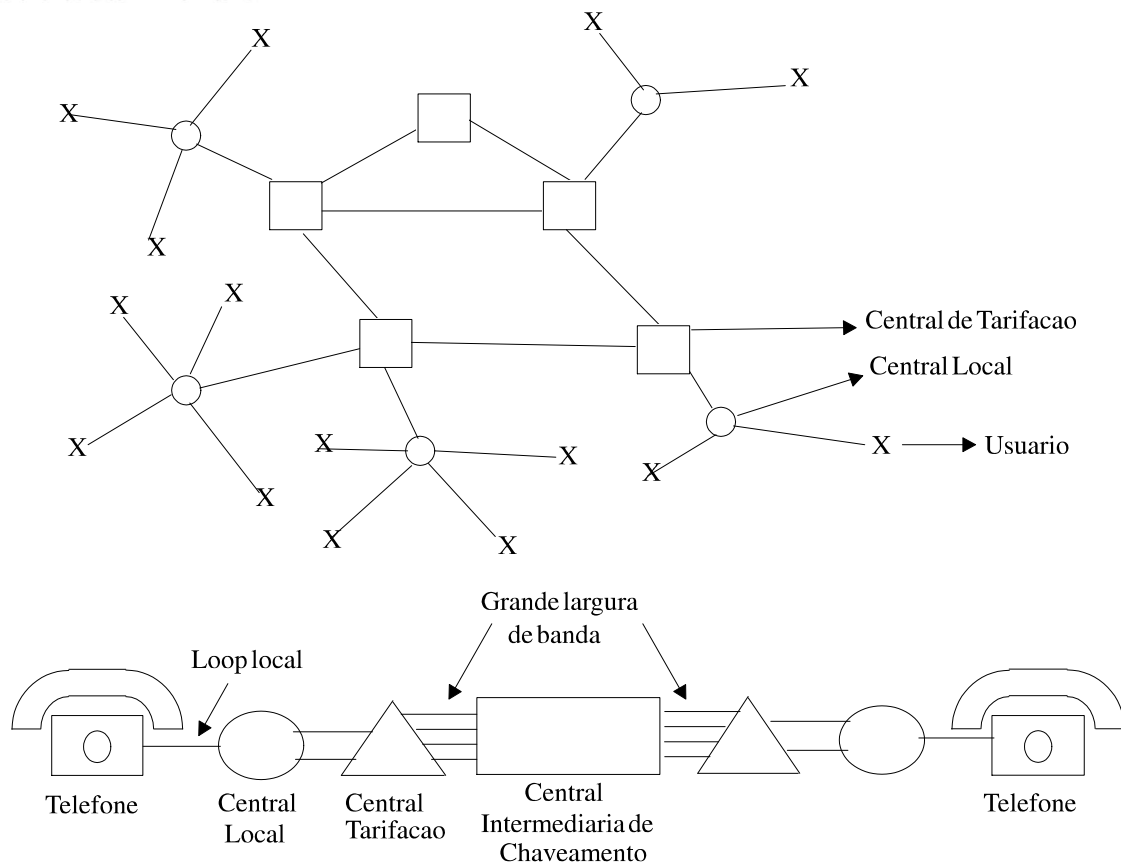
- **Transmissão Analógica: O Sistema Telefônico.**

Conexão direta : 10^7 a 10^8 bps (1 erro em 10^{12})

Linha telefonica : 10^4 bps (1 erro em 10^5)

300 milhoes de telefones instalados.

O sistema telefônico é organizado de maneira altamente redundante com hierarquia de multicamadas.



Modems

As linhas telefônicas normais não podem ser usadas diretamente para interconexão de dois computadores. Os sinais digitais são degradados drasticamente.

MODEM (MODulator DEModulator) converte sinais digitais em analógicos.

"Portadora" ("carrier") : um sinal de 1 a 2 KHz que é introduzido na linha. Sua amplitude, frequência ou fase podem ser modulados para se conseguir transmitir informações.

RS-232C e RS-449

A interface entre o computador e o modem é um exemplo de um protocolo de camada física. Este protocolo deve especificar em detalhes as características mecânicas, elétricas, funcionais e procedurais.



Característica Mecânica - 25 pinos

Características Elétricas - < -3 volts : bit "1"
- > +4 volts : bit "0"
- 24 Kbps (cabos de até 15 metros)

Características Funcionais - O que cada pino significa e quais circuitos são conectados a um determinado pino.

Características Procedurais - Diz qual é a sequência legal de eventos. O protocolo é baseado em pares de ação e reação .

- **Transmissão Digital no Sistema Telefónico**

Vantagens : (1) - Taxa de erros baixa
(2) - Voz, dados, música e imagem ao mesmo tempo
(3) - Taxas de transferencia maiores
(4) - Mais baratos

Metodos de Codificacao:

CODEC (COder-DECoder)

PCM: 8000 amostras por segundo (que permite amostrar sinais de até 4 khz)

T1 (Bell System): 24 canais de voz multiplexados. A cada 125 microssegundos, passam 193 bits. Capacidade de transmissao: 1.544 Mbps

Circuit Switching x Packet Switching

Multiplexação de sinais são importantes para a utilização eficiente de canais de comunicação.

FDM - Frequency Division Multiplexing

TDM - Time Division Multiplexing

Estes métodos são adequados para transmissão de voz. Para dados, métodos diferentes devem ser empregados.

- **Circuit Switching**

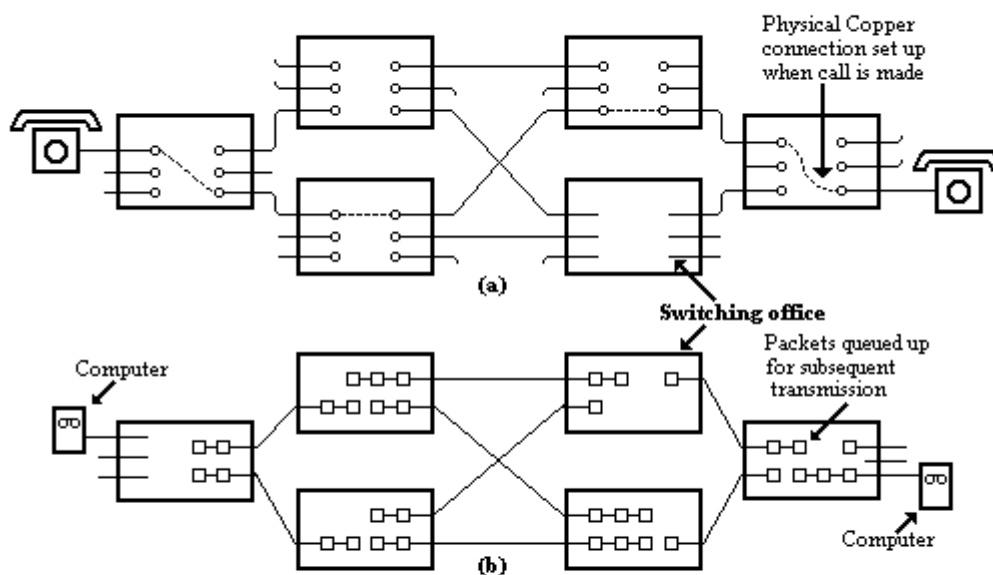
Quando uma conexão é feita, um caminho dedicado é aberto entre a fonte e o destino. Um caminho porta-a-porta deve ser estabelecido antes da transmissão de qualquer dado.

- **Packet Switching**

Os tamanhos de blocos são limitados. Os IMPs não têm que dispor de buffers para armazenar blocos longos."A principal razão para implementação de packet switching é evitar o tempo de conexão.

Circuit Switching - para voz

Packet Switching - para dados



a) Circuit Switching b) Packet Switching

1.2 A Subcamada de Acesso ao Meio

Introdução à Teoria das Filas

Ferramenta básica para análise quantitativa de redes de computadores.

- **Sistemas De Enfileiramento**

Modela processos em que usuários:

- chegam
- esperam
- são atendidos
- saem

Ex.: fila de caixa (supermercados, bancos, etc)
sala de espera de clínicas

Sistemas de filas podem ser caracterizados por 5 componentes:

1. Função densidade de probabilidade de tempo de chegadas
2. Função densidade de probabilidade de tempo de serviços
3. O numero de servidores
4. O método de disciplina da fila
5. A quantidade de espaços de buffer nas filas

A densidade de probabilidade de tempo entre chegadas descreve o intervalo entre chegadas consecutivas.

Para analisar o sistema de fila, o tempo que cada usuário toma do servidor deve ser conhecido. Este tempo varia de usuário para usuário.

O número de servidores também é importante. Por exemplo , em muitos bancos se vê uma grande fila única para todos os clientes (multi-servidor). Em outros, cada caixa tem sua fila própria. Temos aí uma coleção de filas de servidor único.

A disciplina da fila descreve a ordem na qual os usuários são tomados da fila:

- Supermercados e bancos : primeiro que chega, primeiro a ser servido.
- Pronto socorro : primeiro a ser atendido é o caso mais grave.
- Alguns sistemas de fotocópias : trabalhos menores primeiro.

Nem todo sistema de filas tem um espaço de buffer infinito. Quando muitos usuários estão enfileirados, alguns podem ser rejeitados.

Nossa análise se concentrará em sistemas :

- com espaço de buffer infinito
- com um único servidor
- primeiro a chegar, primeiro a ser servido

Para sistemas com a notação A/B/m é utilizada, onde :

A : é a função densidade de probabilidade do tempo entre chegadas.

B : é a função densidade de probabilidade do tempo de serviço

m : é o número de servidores.

As densidades de probabilidades (A e B) são escolhidas entre :

M : exponencial

D : todos os usuários têm o mesmo valor (D de determinístico)

G : genérico

Nós assumiremos o modelo M/M/1, que é razoável para qualquer sistema que tenha um número grande de usuários independentes.

Nestas circunstâncias, a probabilidade de exatamente n usuários chegarem durante um intervalo de duração t é dado pela Lie de Poisson :

$$P_n(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t}$$

onde λ é a velocidade média de chegada.

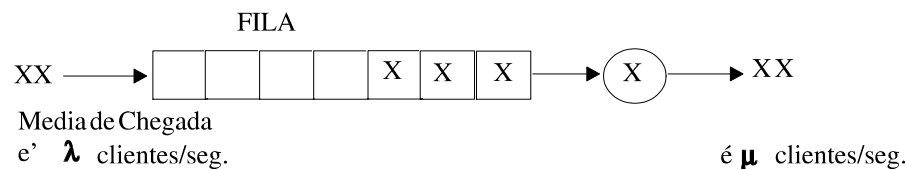
Deste resultado pode-se provar que uma distribuição de tempo entre chegadas definidas pela lei de Poisson gera uma função exponencial de densidade de probabilidades.

$$a(t)dt = \lambda e^{-\lambda t} dt$$

Pode-se também mostrar que se a probabilidade de serviços terminando em algum intervalo Δt é $\mu\Delta t$, então a função densidade de probabilidade para o tempo de serviço é $\mu e^{-\mu t}$ com o tempo médio de serviço de $\frac{1}{\mu}$ segundos por usuário.

• Os Sistemas M/M/1 Em Equilíbrio

O estado de um sistema M/M/1 é completamente descrito quando se define quantos usuários estão correntemente no sistema, incluindo os na fila de espera e aqueles sendo atendidos.



P_k é a probabilidade de que existam exatamente K clientes no sistema (fila + servidor) em equilíbrio.

A partir do cálculo dos P_k s , pode-se achar:

- o número médio de clientes no sistema
- o tempo de espera
- outras estatísticas do sistema

$$P_k = (1 - \rho) \rho^k$$

onde $\rho = \frac{\lambda}{\mu}$

O número médio de clientes no sistema será :

$$N = \frac{\rho}{1 - \rho}$$

O tempo de espera total, incluindo o tempo de atendimento será:

$$T = \frac{1}{\mu - \lambda}$$

- **Redes Com Filas M/M/1**

Com algumas mudanças de notação, o resultado acima pode ser utilizado para resolver o problema de achar o atrazo de enfileamento para packets num IMP.

$$T_i = \frac{1}{\mu C_i - \lambda_i}$$

onde C_i é a capacidade de comunicação do canal i em bits/segundo e μC_i é a taxa de serviço em packets/segundo.

Alocação de Canais

Classes de Rede: ponto-a-ponto
broadcast

Em redes do tipo *broadcast*, a questão central é: quem consegue acesso ao canal (meio) quando existe competição por ele.

Trataremos dos diferentes métodos de solução do problema do controle de acesso ao meio (MAC).

- **Redes Locais e Metropolitanas**

MACs são especialmente importantes em LANs, pois praticamente todas usam canais de múltiplos acessos ao contrário das WANs que utilizam ligações ponto-a-ponto.

Existe uma relação forte entre LANs e canais de múltiplo acesso, de modo que as LANs também serão estudadas.

- **Características de Lans**

1. Um diâmetro não mais que alguns quilômetros.

2. Uma taxa de transmissão total de pelo menos vários Mbps.
3. São controladas por uma única organização.

- **Características de Wans**

1. Se espalham sobre países inteiros
2. Tem taxa de transmissão próximas de 1 Mbps
3. São controladas por múltiplas entidades. (As companhias de telecomunicações possuem a subnet de comunicação e os clientes possuem os hosts).

Entre LANs e as WANs, estão as MANs (Metropolitan Area Networks) que se espalham por cidades inteiras, mas usam tecnologia de LANs.

Projetistas de WANs são sempre forçados (por razões políticas, legais ou econômicas) a usarem a rede telefônica pública apesar dos seus problemas.

Projetistas de LANs podem projetar seus próprios meios de transmissão com a largura de banda desejada.

- **Alocação Estática de Canais**

Quando existe um número de usuários pequeno e fixo e cada usuário tem uma carga grande de tráfego, FDM é um mecanismo simples e eficiente de alocação. Problemas:

- Se a banda tem N slots e nem todos os usuários precisam utilizar o meio, o sistema fica ineficiente.
- Se mais que N precisam se comunicar, alguns não terão permissão, mesmo se alguns dos que tenham conseguido permissão não estejam transmitindo.

- **Alocação Dinâmica De Canais**

Problema da alocação. Considera-se:

1. Modelo **estação**. N estações independentes (computadores ou terminais) cada uma com um programa ou usuário que gera frames para transmissão. Num intervalo de tempo Δt , a probabilidade de um frame ser gerado é $\lambda \Delta t$, onde λ é constante (taxa de chegada de novos frames).

2. Canal único. Apenas um canal é disponível para toda a comunicação.

3. Colisão. Se dois frames são transmitidos simultaneamente, eles se superpõem no tempo e o sinal resultante não contém informação. Todas as estações devem detectar colisões.

4a. Tempo contínuo. Transmissão de frames pode começar a qualquer instante.

4b. Tempo em slots. O tempo é dividido em intervalos discretos.

5a. Detecção de portadora. Estações percebem se o canal está em uso antes de tentar utilizá-lo.

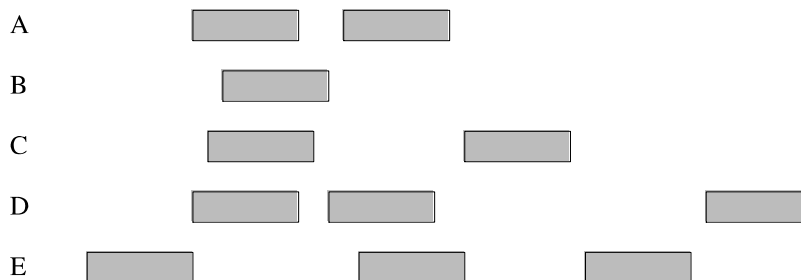
5b. Portadora não detectada. Estações não sabem sobre o status do canal. Apenas transmitem.

Protocolos Aloha

Princípios:

- Deixe os usuários transmitirem sempre que tiverem dados a serem enviados.
- Colisões existirão e os frames serão destruídos. Entretanto, a fonte sempre consegue saber se o frame foi destruído ou não "escutando" o canal de saída.
- Se o frame foi destruído, a fonte espera por um intervalo de tempo aleatório e o reenvia.

Contention Systems: sistemas em que múltiplos usuários compartilham um canal comum de maneira que conflitos podem acontecer.



ALOHA puro:

- dois frames ao mesmo tempo → ambos destruídos
- último bit de um frame coincide com o primeiro bit de outro → ambos destruídos

Qual a eficiência de um canal ALOHA?

Consideremos:

- *frame time*: quantidade de tempo necessário para transmitir um frame de padrão de tamanho fixo.

- Uma população infinita de usuários gera novos frames de acordo com a distribuição de Poisson com média de S frames por frame time.

- Se $S > 1$ a população está gerando frames numa taxa que não pode ser acompanhada pelo canal. Para um desempenho razoável.,

$$0 < S < 1$$

- Assume-se que a probabilidade de K tentativas de transmissão por frame time é também Poisson com média G por frame time. Logo,

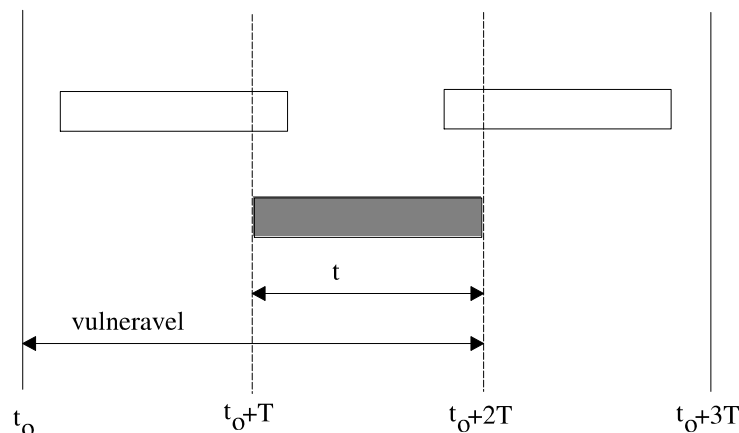
$$G > S$$

- P_0 é a probabilidade de que um frame não sofra colisão.

- O desempenho será a carga G , vezes a probabilidade de que o frame não sofra colisão:

$$S = G P_0$$

Em que condições um frame é transmitido sem danos?



A probabilidade de que K frames sejam gerados durante um frame time é dado pela distribuição de Poisson:

$$\Pr[K] = \frac{G^K e^{-G}}{K!}$$

Para $K=0$ e $P_0 = e^{-G}$

Para um intervalo de dois frames time, a média de frames gerada é $2G$. Logo,

$$P_0 = e^{-2G} \quad \text{e} \quad S = G e^{-2G}$$

O melhor desempenho será quando $G = 0.5 \rightarrow S = 0.184$

Em 1972: SLOTTED ALOHA

Divisão do tempo em intervalos (slots), cada um correspondendo a um frame. Uma estação emite um bip no início de cada intervalo. Um usuário sempre espera o início do próximo slot.

$$S = Ge^{-G}$$

Com melhor desempenho para $G = 1 \rightarrow S = 1/e \cong 40\%$

Em 1985 \rightarrow Slotted Aloha com população finita

Protocolos CSMA

Em LANs, as estações podem monitorar as atividades do canal e escolher o momento certo para tentar acessá-lo.

- **Protocolos com detecção de Portadora**

Protocolos em que as estações examinam o canal para saber se ele está ocupado ou não.

CSMA (*Carrier Sense Multiple Access*)

- 1-persistente:

- 1) Estação verifica se alguém está utilizando o canal.
- 2) Se o canal está ocupado, a estação espera até que ele se torne livre e transmite.
- 3) Se uma colisão ocorre, a estação espera por um tempo aleatório e começa tudo novamente.

- Não-persistente:

- 1) A estação verifica se alguém está utilizando o canal.
- 2) Se desocupado, envia.
- 3) Se o canal está ocupado, a estação não permanece monitorando a linha. Ela espera por um tempo aleatório e repete o algoritmo.

- P-persistente: (para slotted channels)

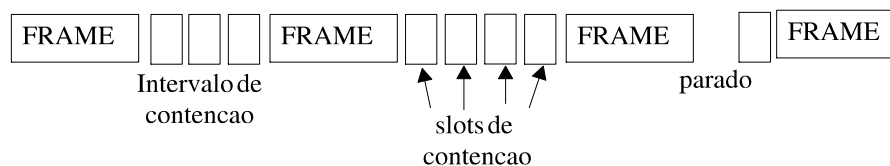
- 1) A estação monitora o canal.
- 2) Se o canal está desocupado, a estação transmite com probabilidade p . Com uma probabilidade $q = 1-p$, ela desiste até o próximo slot.
- 3) Se aquele slot também está desocupado, a estação ou transmite ou desiste de novo, com probabilidade p e q .

O processo é repetido até que o frame seja transmitido ou outra estação tenha começado a transmitir.

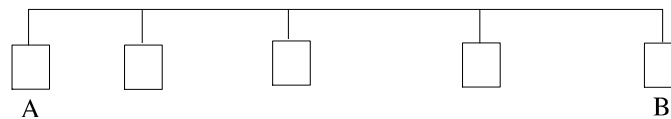
- **CSMA com detecção de colisão**

No caso de haver colisão entre frames transmitidos por duas estações, as duas devem parar imediatamente a transmissão.

Modelo conceitual do CSMA/CD



Quanto tempo é necessário para se perceber uma colisão? (Isto é, qual o tempo do contention slot?)



Tempo do percurso $A \rightarrow B = \Psi$

Pior caso: (1) A começa

(2) Num tempo $\Psi - \varepsilon$ B começa e percebe a colisão. Pára.

(3) O efeito da colisão chega a A num tempo $2\Psi - \varepsilon$

Portanto 2Ψ é o tempo necessário para que a estação esteja segura que assumiu o controle. Num cabo de 1 Km, $\Psi = 5$ microsegundos.

- **Protocolos sem colisões**

Embora colisões não ocorram durante a transmissão de um frame numa rede CSMA/CD, a partir do momento em que uma estação assume o canal, elas ainda podem ocorrer durante o intervalo de contenção.

Considere N estações com endereços de 0 a N-1.

Método Bit-Map Básico:

- Cada *contention period* consiste de N slots.
- Se uma estação j tem frame pronto, ela transmite um bit 1 no slot número j.
- Depois que todos os slots passarem, cada estação tem conhecimento de quais outras estações querem utilizar o meio e começa a transmitir em ordem.
- Depois que todas estações transmitem, começa outro período de N bits de contenção.

BRAP - Broadcast Recognition with Alternating Priorities

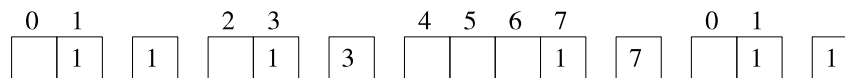
O bit-map básico tem alguns problemas graves:

- 1) Estações com número alto têm melhor serviço que as de número baixo.
- 2) A baixa carga, uma estação deve esperar o final da varredura corrente para transmitir.

BRAP resolve ambos os problemas:

Quando uma estação coloca um bit 1 no seu slot, ela começa a transmissão do frame imediatamente.

Ao invés de iniciar a varredura com a estação zero toda vez, ela é reiniciada com a estação seguindo a que acaba de transmitir.



Padrão IEEE 802 para LANs

802.1 - Descrição dos Protocolos

802.2 - Descrição da camada Link Logic Controls

802.3 - Ethernet (CSMA/CD) 1-persistente

802.4 - Token Bus

802.5 - Token Ring

- **Ethernet**

Padrão: Cabo coaxial grosso (thick) de 50 ohms (10B5)

Outros meios: Cabo coaxial fino (thin) - também chamado de cheapernet. (10B2)

Par Trançado (10BT)

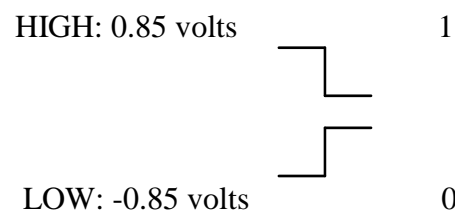
Fibra Optica (10BF)

Thin - usa junções T

Thick - usa TAP's

Par Trançado - Usa Hubs

Os sinais são codificados: (Manchester Encoding)



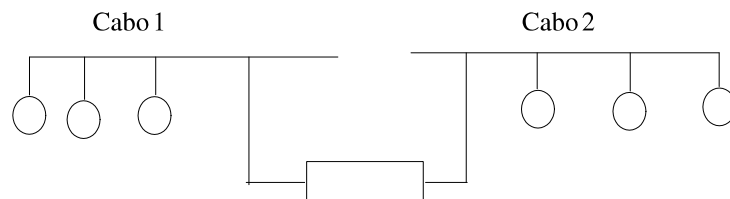
Transceiver: circuitos para detectar transmissão e colisão.

Cabo do transceiver: conecta o transceiver à interface do computador.

Interface: Monta os dados no frame apropriado. Computa *checksum* nos frames de saída e verifica nos frames de entrada. Implementa *buffering* para frames de entrada. Implementa *queueing* para frames de saída.

Maior comprimento do cabo: 500 metros.

Para maiores distâncias é necessário o uso de repetidores.



Comprimento máximo: 2.5 Km.

Ethernet (Subcamada MAC)

7	1	6	6	2	0-1500	0-46	4
Preâmbulo	Iní- cio	Endereço Destino	Endereço Origem	Tam.	Dados	PAD	CRC

Preâmbulo: 7 bytes 10101010

Start of frame: 10101011

Destination address e Source address: 2 ou 6 bytes (para 10 Mbps)

Multicast: envio de uma mesma mensagem para um grupo de estações.
(MSB = 1)

Broadcast: envio de uma mesma mensagem para todas as estações.
(Todos os bits = 1)

Lenght: dá o comprimento do *data field* (0-1500). Para evitar problemas, o tamanho do frame que vai do destination address até o checksum deve ser maior ou igual a 64 bytes.

Qualquer estação detetando uma colisão aborta sua transmissão e gera um ruído para prevenir todas as outras estações e então espera um tempo aleatório antes de repetir o ciclo novamente.

Depois da primeira colisão, cada estação espera 0 ou 1 slot time antes de tentar novamente. Depois de uma segunda colisão, a espera será de 0, 1 ou 2 ou 3 slot times. Numa terceira colisão, a espera será entre 0 e 7. E assim por diante.

Depois de 16 colisões, uma falha é relatada aos protocolos superiores.

Performance

Seja A a probabilidade de alguma estação alocar o ETHER durante um slot

$$A = Kp(1-p)^{K-1}$$

onde K: número máximo de estações prontas para transmitir.
p: probabilidade de uma estação transmitir durante um slot.

$$\text{Eficiência} = \frac{p}{p + 2\Psi/A}$$

p: tempo médio de transmissão de um frame.

Ou,

$$\text{Eficiência} = \frac{1}{1 + 2BLe/CF}$$

onde,

B = largura de banda

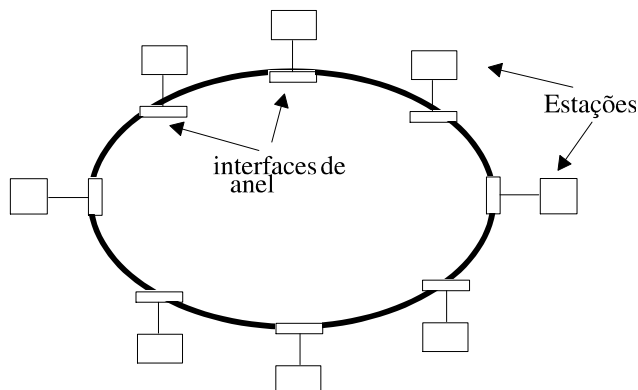
L = comprimento do cabo

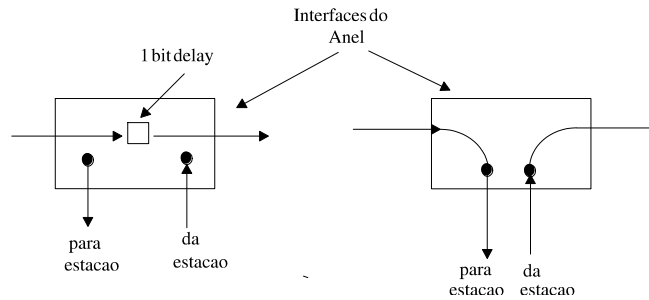
C = velocidade de propagação

F = comprimento do frame

• Token Ring (IEEE 802.5)

- Não é broadcast (de fato).
- Coleção de links ponto a ponto formando um círculo.
- Pode rodar em par trançado, cabo coaxial ou fibra ótica.
- É completamente digital.
- É apropriado para tempo real.
- Quando a rede não está ocupada um token circula entre as estações.
- Não existem colisões.





"Comprimento físico" de um bit.

Quanto um bit ocupada dentro de uma linha de transmissão?

Se temos um bit rate de x bps, isto significa que 1 bit é transmitido cada $1/x$ segundos.

Se temos um bit rate de x Mbps, isto significa que 1 bit é transmitido a cada $1/x$ microsegundos.

A velocidade de propagação de um sinal elétrico num cabo coaxial é da ordem de 200 metros por microsegundo.

$$D = V \cdot T$$

$$D = 200 \text{ m/microseg.} \cdot 1/x \text{ microseg.}$$

$$D = 200/x \text{ metros}$$

Por exemplo, numa rede rodando a 10 Mbps: $D = 20$ metros.

Regras de Acesso

Quando uma estação tem um frame para ser transmitido, ela deve se apossar do token e removê-lo do anel, antes de transmitir.

O ring deve ter um delay suficiente para conter um token completo quando todas as estações estão desocupadas.

Modos:

A-Escuta: bits na entrada são copiados na saída (1 bit delay)

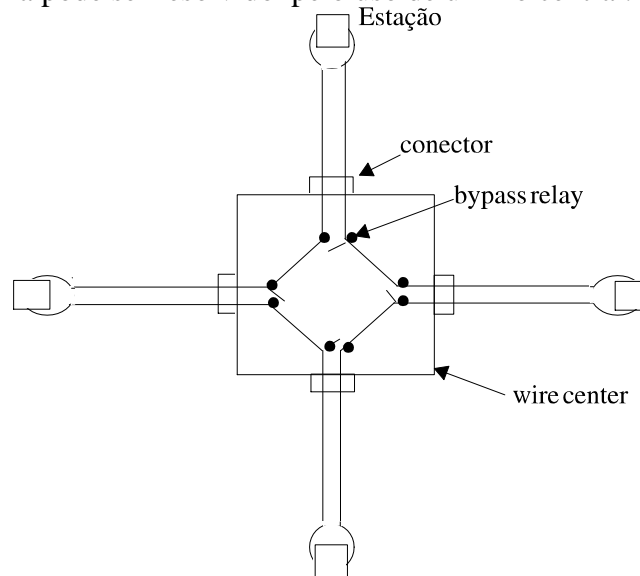
B-Transmite: (depois de possuir o token) A estação quebra a conexão entre entrada e saída, colocando o seu próprio dado no anel.

Bits que chegam depois de circular pelo anel são retirados pela estação que os envia.

Assim que uma estação recebe de volta seu último bit ela deve chavear para o modo escuta e regenerar o token.

Uma crítica aos rings é a confiabilidade: se uma estação quebra o anel cai.

Este problema pode ser resolvido pelo uso de um fio central.

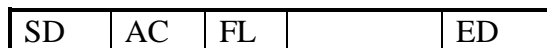


Os relays são alimentados por correntes oriundas das estações. (*Star shaped ring*).

Formato do frame token:



Formato do frame de dados:



Em condições normais o primeiro bit do frame vai circular todo o anel e retornar antes de terminar a transmissão do frame. Por isto, a estação retransmissora deve retirar os bits que ela coloca na rede.

Cada estação tem um tempo determinado máximo durante o qual ela pode reter o token. Se, transmitido o primeiro frame, ainda restar tempo, extra frames podem ser enviados.

O anel deve apresentar um delay suficientemente grande para conter o token.

Delays artificiais podem ser inseridos.

Acknowledgements são feitos pela inversão de um bit do frame lido.

Meio físico:

- Par trançado (blindado ou não)
- Cabo Coaxial.
- Fibra ótica

Manutenção do ring

Enquanto o controle das redes em duto é feito de maneira descentralizada, Token Ring tem uma estação de monitoramento.

Qualquer estação tem capacidade de ser monitora. Tarefas:

- cuidar para que o token não se perca.
- tomar providências quando o ring quebrar.
- limpar o ring quando frames danificados aparecem.
- tomar providências relacionadas a frames orfãos.

Redes de Fibra Ótica

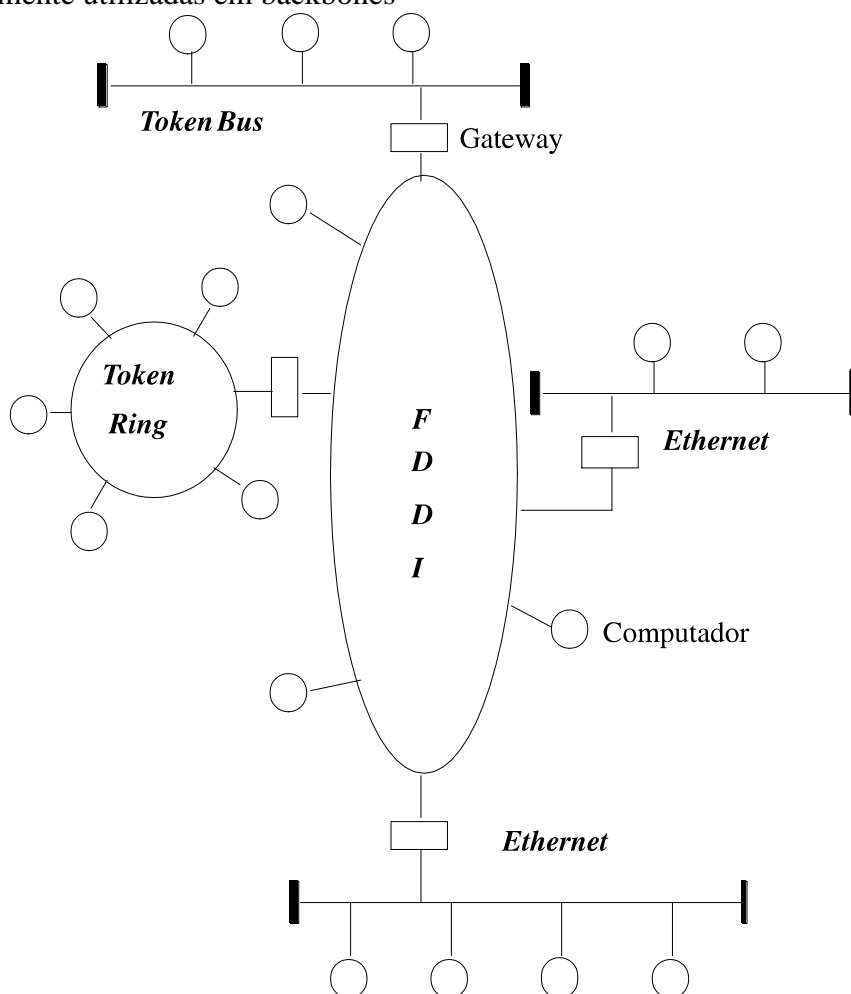
Vantagens:

- Alta largura de banda.
- É fina e leve.
- Não afetada por ruídos eletromagnéticos oriundos de relâmpagos ou aparelhos eletrônicos.
- muito segura, difícil fazer "grampeagem" sem ser detetado.

- **FDDI (Fiber Distributed Data Interface)**

LAN TOKEN RING:

- rodando a 100 Mbps.
- distâncias de até 200 Km.
- até 1000 estações conectadas.
- normalmente utilizadas em backbones



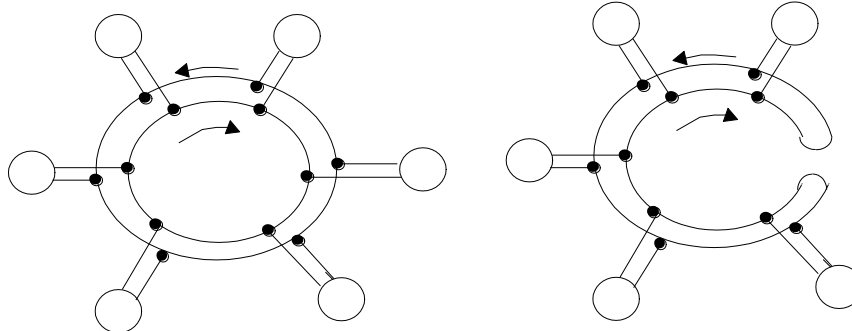
Usa multimode fiber

Usa LED ao invés de laser

Consiste de dois canais em fibra; um no sentido horário e outro no sentido anti-horário.

Usa fibra multimode (não é necessário usar single mode para transmitir a 100 Mbps).

1 erro em 2.5×10^{10} bits (pior caso)



Se um anel quebra, o outro assume.

Se os dois quebram no mesmo ponto, o anel é reconfigurado.

Utiliza-se Central Wire.

Duas classes de estações:

DAS: Se conectam aos dois rings

SAS: Se conectam a apenas um ring

Transmissão: (como no 802.5)

1. Captura do token
2. Transmissão
3. Remoção do frame.

A estação deve regenerar o token logo após a transmissão de seu frame (diferente do 802.5).

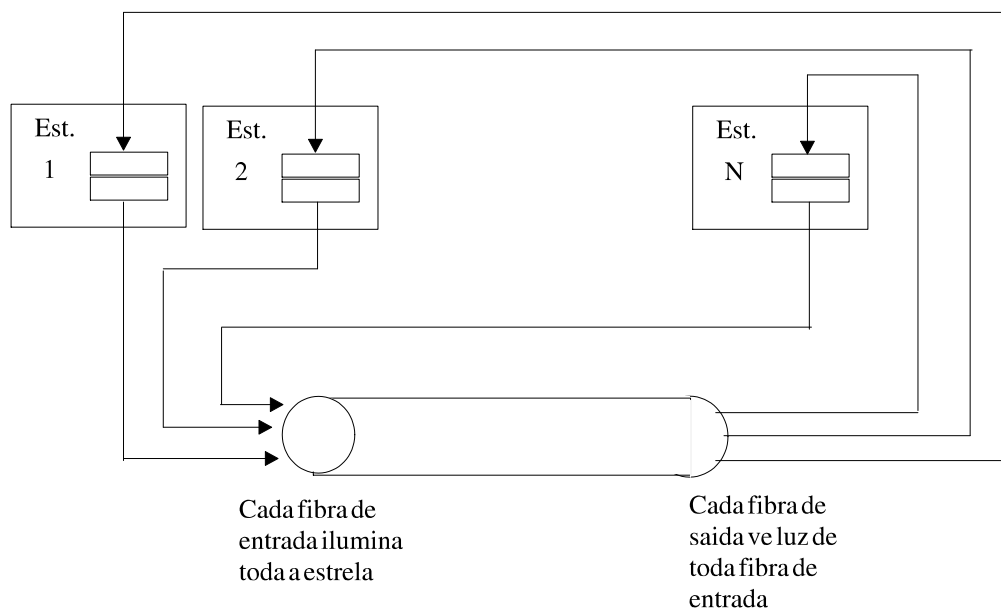
O formato dos frames do FDDI são similares ao 802.5.

Adicionalmente, permite a transmissão síncrona de frames para uso em transmissão de voz (PCM) e tráfego ISDN.

• Fibernet II

Objetivo: Construir uma rede de fibra ótica compatível com a Ethernet a nível de transceiver.

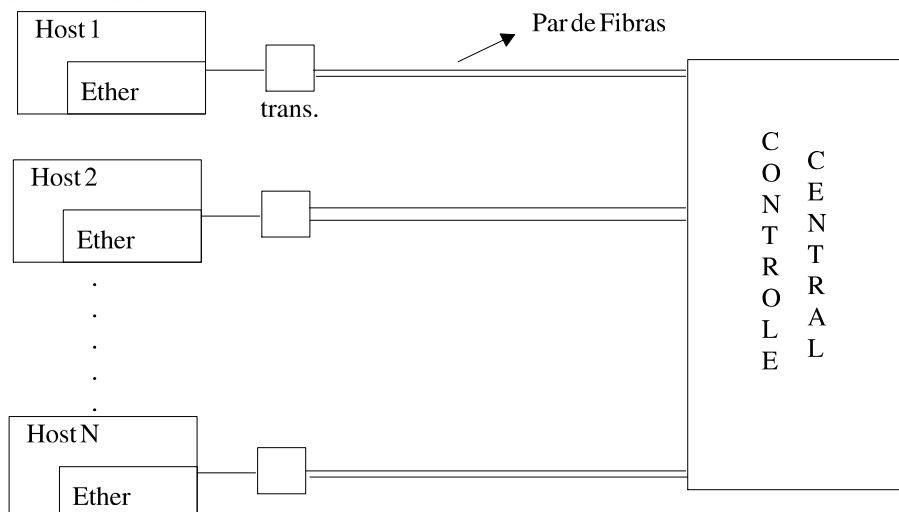
Problema: No desenvolvimento de CSMA/CD sobre fibra, como detectar colisões?



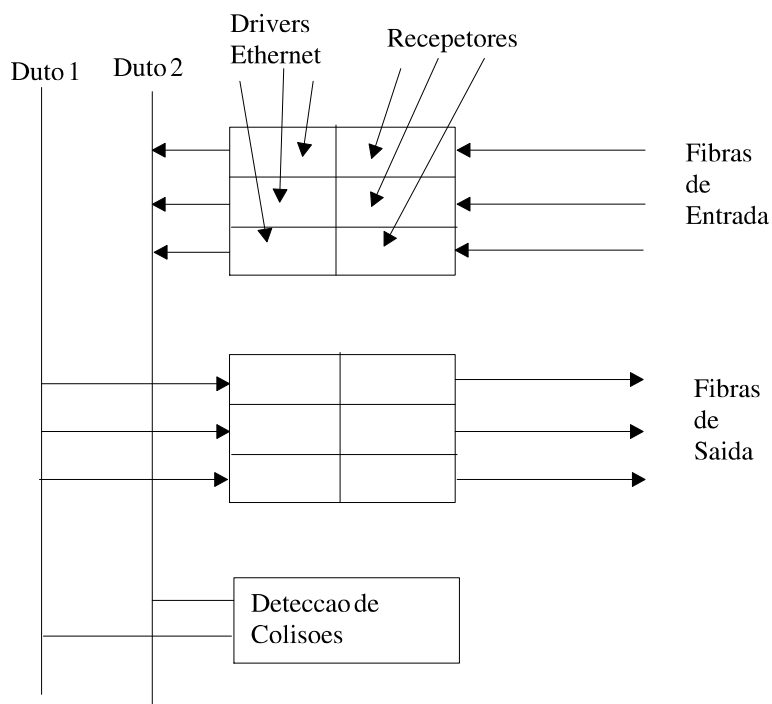
Métodos:

1. Detecção por medida de potência. Se uma estação "enxerga" mais potência do que ela está colocando no meio, é porque houve colisão.
2. Comprimento do pulso. Se duas estações colidem, o pulso que é "sentido" será provavelmente mais que o transmitido.
3. Tempo de atraso. Quando duas estações colidem, a que transmitir por último vai receber o sinal da primeira antes que o próprio sinal possa ser recebido. A diferença pode ser detectada.
4. Acoplamento direcional. Projeta-se o receiver de modo que ele não receba sua própria emissão. Se alguma luz é sentida durante a transmissão, ela será devido a colisões.

Como a Passive Star enfraquece o sinal, a Fibernet II usa uma estrela ativa.



No controle central:



Se ocorre uma colisão, esta é detectada do modo usual e as estações informadas.

Se não ocorre colisão, o sinal é difundido para todas as saídas.

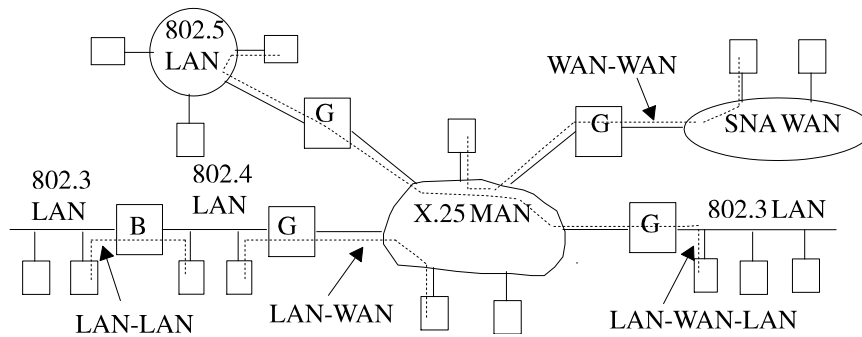
PARTE 3 - Interconexão de Redes

Internet - interconexão de duas ou mais redes.

Como existem muitos tipos de redes rodando protocolos diferentes, o problema de interconexão tem ganhado muita atenção.

Tipos de tráfego internet:

1. LAN-LAN
2. LAN-WAN
3. WAN-WAN
4. LAN-WAN-LAN

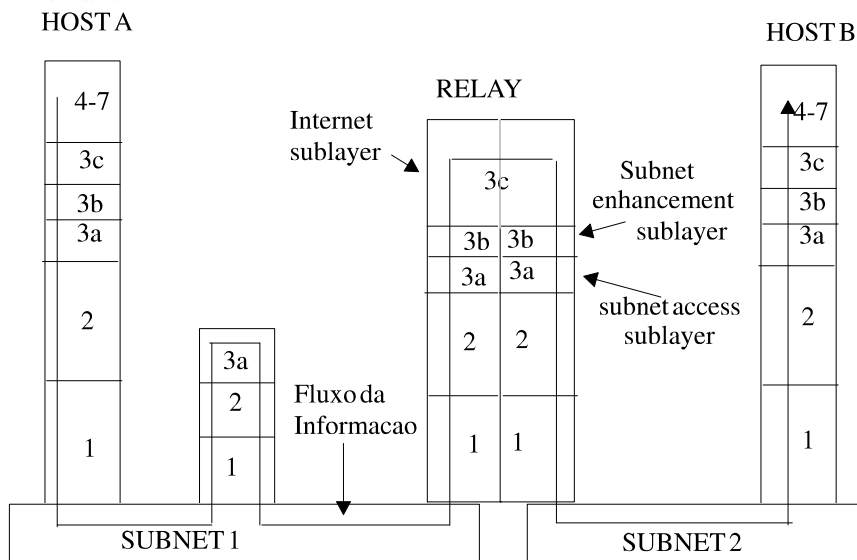


Relays - gerencia conversões quando um pacote anda de uma rede para outra.

No modelo OSI: Internetwork é feito a nível de Network Layer.

A camada Network pode ser subdividida em três subcamadas:

- Subnet Access sublayer: gerencia o protocolo de camada Network para a subnet específica.
- Subnet Enhancement sublayer: projetado para compatibilizar subnets que oferecem serviços diferentes.
- Internet Sublayer: responsável por roteamento end-to-end. Quando um pacote chega a um relay, ele deve subir até o internet sublayer.



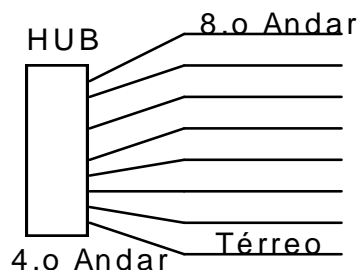
Tipos de Relays:

- **Layer 1:** Repetidores e Concentradores. Copiam bits individuais entre dois segmentos de cabo (repetidor) ou distribui sinais para estações ligadas em estrela (concentrador). São dispositivos de baixo nível que amplificam sinais elétricos. Repetidores, Concentradores e, às vezes bridges são chamados **HUBS**.

Ex: Num prédio de 8 andares, poder-se-ia fazer um cabeamento do tipo espinha dorsal ou em estrela, utilizando um repetidor multiportas.



Opção utilizando repetidores Individuais



Opção usando um repetidor Multiportas

O HUB utilizado acima tem 8 portas AUI. Poderia também ter portas BNC ou FOIRL. Outro tipo de HUB bastante utilizado é o concentrador de portas ethernet Par Trançado.

Seu uso torna as tarefas de projeto e manutenção da rede bastante simplificadas.

Na verdade, um hub ethernet contém uma circuitaria eletrônica de modo a trazer o duto para dentro da caixa. Desta forma, todas as estações estão conectadas diretamente na caixa formando um star-shaped-bus. A grande vantagem do esquema é a facilidade de manutenção e gerenciamento. Hubs ethernet modernos proveem a capacidade de controle

por porta, evitando, inclusive que estações possam acessar dados que não lhe sejam endereçados (em modo promíscuo)

- **Layer 2:** Bridge. Armazena e re-envia frames entre LANs. Recebe um frame e passa à Data Link onde o checksum é verificado. É passado de volta à camada física para ser enviado a uma subnet diferente.
- **Layer 3:** Gateways. Armazena e re-envia pacotes entre redes diferentes. São também chamados de Routers. Redes interconectados por Gateways podem diferir muito mais que aquelas interconectadas por bridges.
- **Layer 4:** Protocol Converters. Fornecem interfaceamento em camadas mais altas.

Repetidores

Podem ser utilizados para:

- * Estender a rede à distâncias maiores que 187 ou 500 metros (no caso de ethernet BNC (cabo fino) e AUI (cabo grosso);
- * Implementar topologias em estrela, como na figura acima.

Switches Ethernet

Os grandes problemas da tecnologia ethernet, oriundos do fato de ser duto e ter o controle de acesso distribuído, as colisões, a susceptibilidade à ruídos eletromagnéticos e a segurança (hacking). A tecnologia de par trançado resolveu em parte estes problemas: a imunidade a ruído e' muito boa, e alguns hubs mais modernos isolam o tráfego por estações. Entretanto, o problema das colisões continuam.

Os inventores das switches exploraram exatamente este problema. Uma switch nada mais é do que um hub ethernet em que o acesso, por porta, é controlado, de modo que as estações não colocam seus dados de forma (persistente). Existem buffers para cada porta, e o acesso é então feito de forma organizada. Além disso, as transferências entre as portas são feitas à velocidades muito superiores que os 10 Mbps, de forma que uma switch garante a velocidade máxima para cada porta.

Dois pontos adicionais completam esta tecnologia que está revolucionando os projetos de redes:

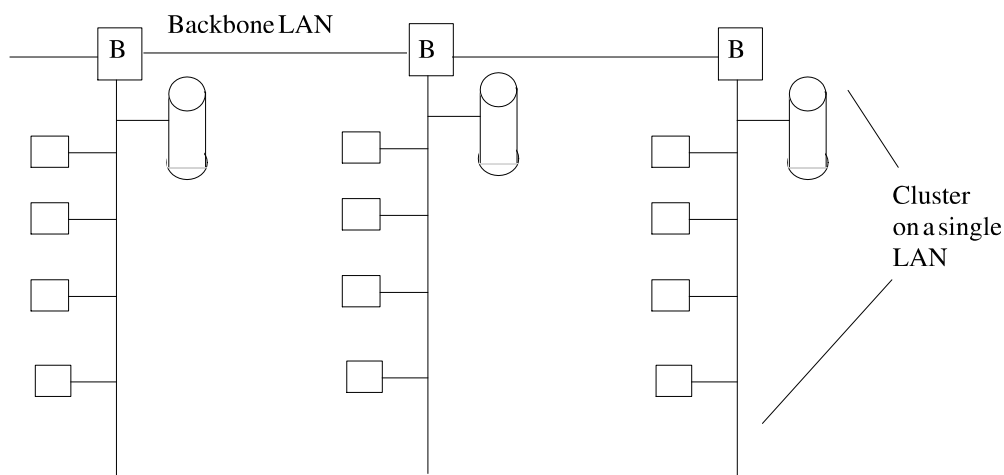
- * tem uma porta de alta velocidade para conexão com servidores;

* cada porta pode ser utilizada por uma estação ou por um grupo de estações (segmento)

Bridges

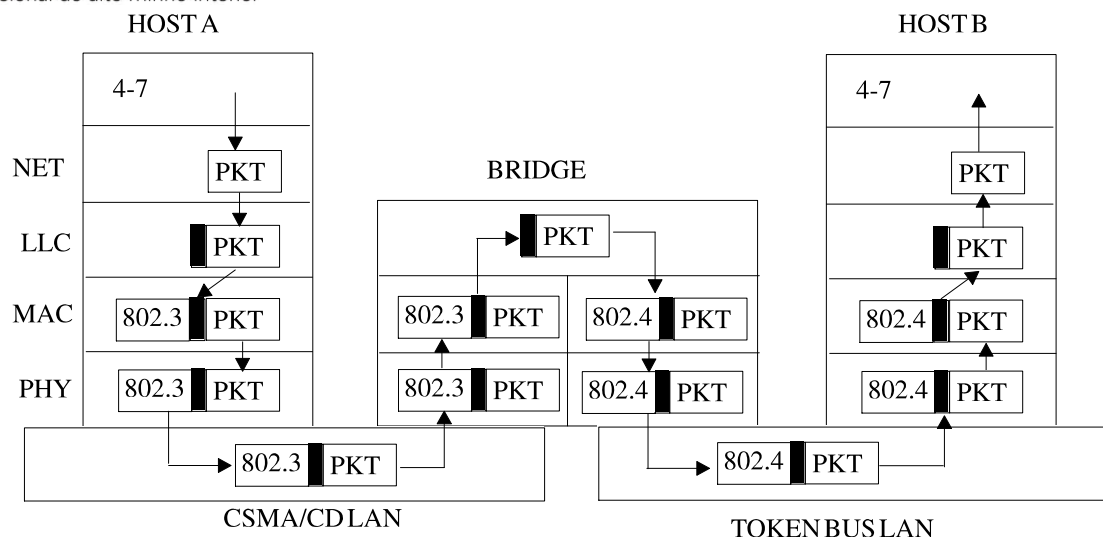
Onde bridges são necessárias:

1. Para dar autonomia a diferentes subnets que querem se comunicar
2. Para interconectar subnets localizadas em áreas geograficamente espalhadas.
3. Para acomodar carga.



4. Para conectar poucas, mas distantes estações.
5. Por confiabilidade. Bridges podem ser colocadas em lugares críticos.
6. Segurança. Colocando bridges, pode-se isolar redes que contenham informações sensíveis.

Note que, com o surgimento das switches, as bridges e os roteadores perderam grande parte de suas funções.



Problemas conectando 802.x e 802.y

1. Diferentes formatos de dados:

- Preâmbulo
- Controle de frame
- Comprimento de frame
- Delimitador de fim

2. LANs interligadas nem sempre tem a mesma velocidade

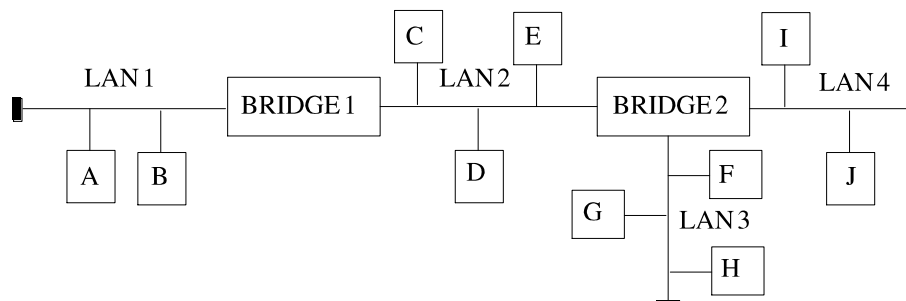
- 802.3 - 10 Mbps (?)
- 802.4 - 10 Mbps (?)
- 802.5 - 4 Mbps

3. Cada LAN tem um comprimento máximo para o frame

- 802.3 - depende da configuração (1518 bytes)
- 802.4 - 8191 bytes
- 802.5 - não existe limite no tamanho, mas no tempo (5000 bytes para 10 microseg.)

Bridges Transparentes

Adotadas por (802.3 e 802.4)



Cada bridge tem uma tabela listando cada possível destino, e a qual linha de saída pertence. Na bridge 2, a estação A aparece na lista de LAN 2.

Inicialmente as tabelas estão vazias. Quando um frame chega para um destino desconhecido, ele é enviado para todas as possíveis saídas e o algoritmo usado para se preencher as tabelas é o backward learning

Principal característica: fácil utilização.

Gateways (ou roteadores)

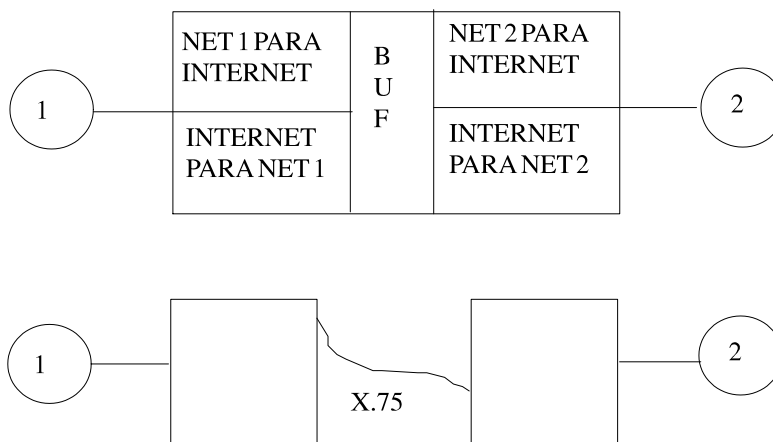
Operam a nível de Network.

Comumente usados em WANs.

Connection-oriented

Problemas ocorrem quando um gateway conecta duas WANs pertencentes a organizações diferentes.

O gateway é dividido em dois e conectado por uma linha qualquer.



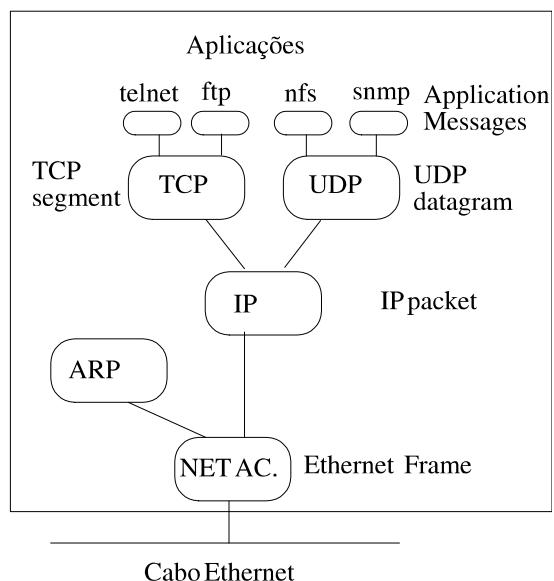
Parte 3. Introdução aos Protocolos TCP/IP

Livro referência: Internetworking with TCP/IP, vol. I, Douglas Comer, Prentice-Hall 1991.

3.1 Conceitos Básicos

Arquitetura Internet

Para começar entender a arquitetura Internet é necessário entender a estrutura:



Internet:

- Interconexão de várias redes físicas diferentes fazendo-as funcionar como uma unidade coordenada.
- Esconde os detalhes de hardware e permite que computadores se comuniquem independentemente de sua rede de comunicação.
- Define detalhes de como computadores devem ser interconectados e uma série de convenções para interconexão de redes e roteamento de tráfego. Aplicável a LAN's, MAN's e WAN's.

Serviços

Ao Nível de Aplicação

Do ponto de vista do usuário, a Internet se apresenta como uma série de programas aplicativos que usa a rede para executar algumas tarefas úteis de comunicação.

- Correio Eletrônico: serviço confiável onde a máquina FONTE se conecta diretamente à máquina DESTINO para entregar as mensagens (normalmente textos curtos).
- Transferência de Arquivos: permite envio e recebimento de arquivos, de programas ou dados de tamanho arbitrário.
- Sessão Remota: permite o usuário de um computador conectar a uma máquina remota e estabelecer uma sessão interativa.

Ao Nível de Transporte

Um programador que escreve programas aplicativos que usam a Internet tem uma visão completamente diferente da do usuário:

- Serviço de entrega de pacotes *connectionless*

A Internet roteia pequenos pacotes de uma máquina para outra baseada nas informações de endereçamento contidas nas mensagens. Ele não garante a entrega. É extremamente eficiente.

- Serviço de Transporte Confiável

Muitas aplicações necessitam muito mais do que a entrega de pacotes, porque condições tais como erros de transmissão, perda de pacotes, ou quedas de IMP's no meio do caminho devem ser checados.

Na Internet: A complexidade está na camada de transporte. A camada de Network é simples e oferece apenas serviços *connectionless*.

Endereçamento

Classes de Endereço:

Endereços Internet

- Internet é uma rede virtual construída pela interconexão de redes físicas através de gateways.
- Um sistema fornece serviços de comunicação universal se ele permite qualquer host se comunicar com qualquer outro. Para efetivar esta universalidade, precisa-se de um sistema de endereçamento globalmente aceito.
- Identificador de hosts:

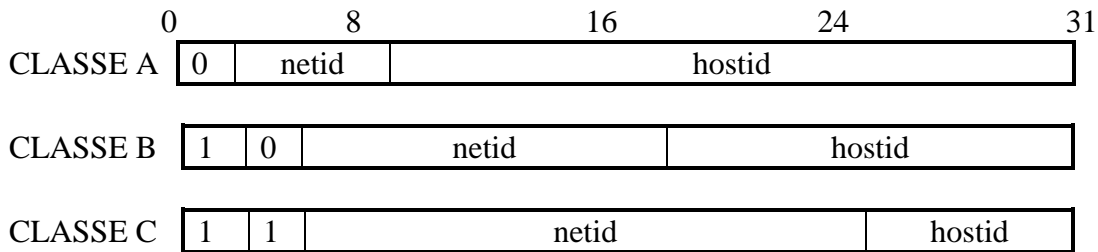
Nome: Qual é o objeto.

Endereço: Onde está o objeto.

Rota: Como chegar ao objeto.

As Três Classes Primárias de Endereços Internet

- A cada host numa rede TCP/IP é atribuído um endereço Internet de 32 bits que é universalmente único e é utilizado para toda comunicação com aquele host.
- Os inteiros componentes do endereço são cuidadosamente escolhidos para fazer, inclusive, o processo de roteamento eficiente.
- Os bits de endereço para todos os hosts de uma determinada rede compartilham um prefixo comum.



Netid: Identifica uma rede.

Hostid: Identifica um host

Class A → 128 redes com até 2^{24} hosts

Class B → 16K redes com até 64K hosts

Class C → 2^{22} redes com até 256 hosts

Um gateway conectando n networks tem n endereços Internet distintos, um para cada conexão.

Notação Decimal com Pontos

Normalmente é usada a notação decimal com pontos para representar os 32 bits de endereçamento. Exemplo:

128.10.2.30

Qual é o hostid?

Mapeamento entre Endereços Físicos e Endereços Internet

Address Resolution Protocol

Se o hospedeiro X quer enviar um pacote IP para o hospedeiro Y:

- X faz um broadcast da mensagem : Onde está Y ?
- *Todos* hospedeiros recebem a mensagem
- *Apenas* hospedeiro Y responde: hospedeiro Y tem ethernet address E.
- X mantém uma cópia do replay
- X envia o pacote para Y, com ethernet address E.

IP sobre ETHERNET

Redes ethernet usam:

- 6 bytes para endereço fonte e destino

- 2 bytes definindo o tipo em cada pacote para permitir múltiplas redes (TCP/IP, Decnet, OSI) num mesmo cabo.
- Data (46 a 1500 bytes)
- 32 bit CRC

Problema

IP : passa Internet address para link de dados.

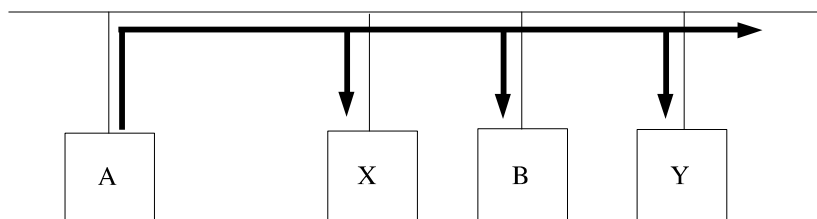
Link de Dados: precisa do Ethernet address para transmitir um frame.

Soluções

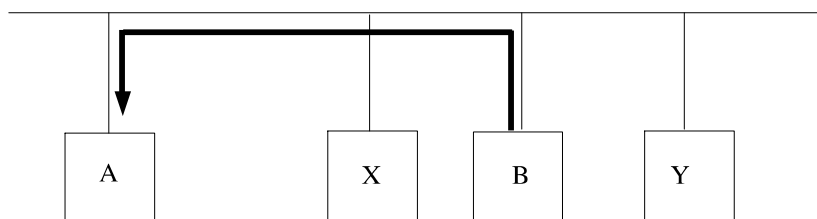
- Tabelas de mapeamento (inconveniente)
- Trocar ethernet address fisicamente (nem sempre possível)
- Utilizar protocolos dinâmicos para descobrir endereços ethernet quando necessitados

Address Resolution Protocol

A quer se comunicar com B

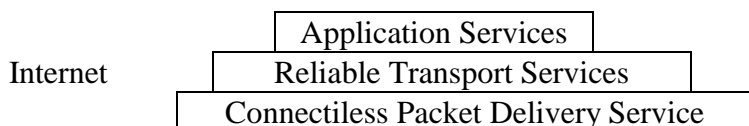


(a) A faz broadcast



(b) só B responde

Protocolo IP



IP define a unidade básica de transferência de dados e o formato exato de dados passados pela Internet. Também inclui uma série de regras especificando como os packets devem ser processados, como os erros devem ser manipulados e contém informações sobre roteamento.

DATAGRAMA: Unidade básica de dados; contém header e área de dados.

0	3	4	7	8	10	12	14	16	19	20	22	24	27	28	31
version		IHL		type of service				total length							
identification								flags		fragment offset					
time to live				protocol				header checksum							
source IP address															
destinaion IP address															
options												padding			
user-data															

VERS: Versão do Protocolo

LEN: Dá o comprimento do header medido em palavras de 32 bits.

TOTAL LENGTH: Dá o tamanho do datagrama medido em bytes.

TYPE OF SERVICE: Especifica como o datagrama deve ser manuseado

PRECEDENCE	D	T	R	NÃO USADO
------------	---	---	---	-----------

Precedence permite especificar a importância do datagrama.

D: pede atraso pequeno

T: pede alta performance

R: pede alta confiabilidade

Fragmentação e remontagem são controladas pelos campos IDENT, FLAGS e FRAGMENT OFFSET.

IDENT: Contém um inteiro que identifica o datagrama. Todo gateway que fragmenta o datagrama, copia o IDENT em cada um dos fragmentos.

FLAGS: Controla a fragmentação. DO NOT fragment. More fragments.

FRAGMENT OFFSET: Especifica o offset deste datagrama no datagrama original em múltiplos de 8 bytes.

TIME: Especifica quanto tempo em segundos o datagrama deve permanecer dentro da Internet.

PROTO: Especifica o formato e conteúdo dos dados pela identificação do protocolo de alto nível.

HEADER CHECKSUM: Garante a integridade dos valores do header.

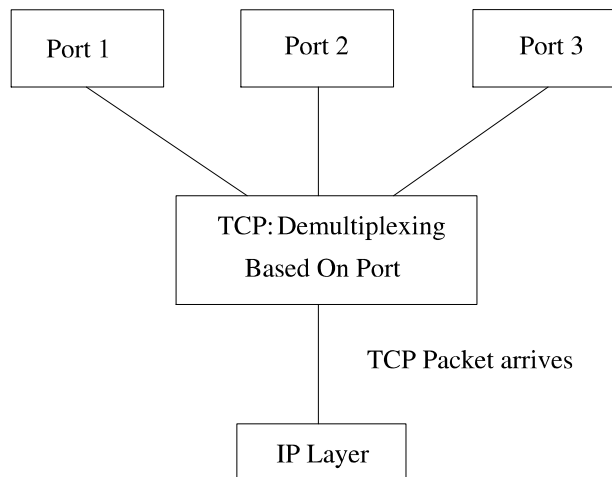
DATA: Início da área de dados.

PADDING: Usado para garantir que o header tenha um tamanho múltiplo de 32 bits.

OPTIONS: Para testes e debugging da rede.

O Protocolo TCP (Transmission Control Protocol)

- TCP é o protocolo confiável para transmissão de *byte streams* de dados.
 - Multiplexado
 - Sequenciado (bytes chegam na ordem correta)
 - Implementa controle de Fluxo (máquina destino pode "frear" a fonte)
- *Byte Stream* é implementado pelo empacotamento conjunto de informações de tamanho arbitrário
- **Multiplexação**



• Header TCP

0		8		16		31	
Souce Port				Destination Port			
Sequence Number							
Acknowledgement							
Off.		Res.	Code		Window		
Checksum				Urgent Pointer			
Options						Padding	
Data							
.....							

- Multiplexação (*Source Port, Destination Port*)
- Sequenciamento (*Sequence number*)
- Confiabilidade (*Acknowledgement Number*). Qualificado por ACK.
- Controle de Fluxo (*Window*)

• Confiabilidade

Pergunta: Como pode um protocolo oferecer transferência confiável se o sistema utilizado na camada inferior apenas oferece transferência não confiável ?

Resposta:

- Aviso de recebimento positivo (*Positive acknowledge*)

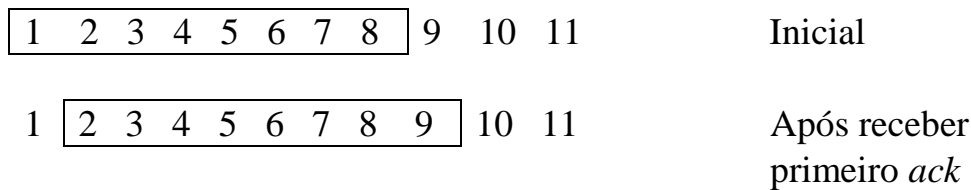
- Retransmissão
- Mais *Sliding Window* (para aumentar a eficiência)

O *Sender*:

- Envia dado (mantem cópia)
- Ativa *timer*
- Retransmite se alcança *timeout* sem ter recebido *ack*.

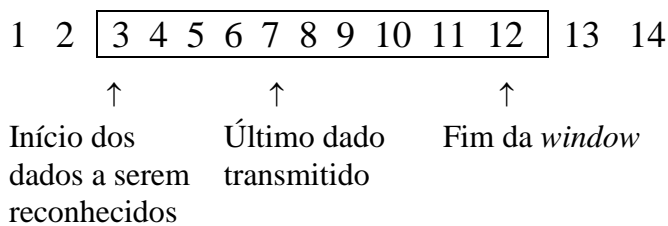
• Sliding Windows:

Evita que o *sender* fique esperando o *acknowledge* para transmitir o próximo pacote.



Um timer é mantido para cada pacote enviado.

No TCP: 3 ponteiros.



- Implementa *sliding windows* ao nível de byte.

Confiabilidade no TCP

- Destinatário acusa recebimento de "cada byte"
- Na realidade, o reconhecimento é feito do "último byte recebido + 1"
- Exemplo

Fonte:

42 ↓ A b c d e f	48 ↓ g h i j k l m n
------------------------	----------------------------

Destinatário:

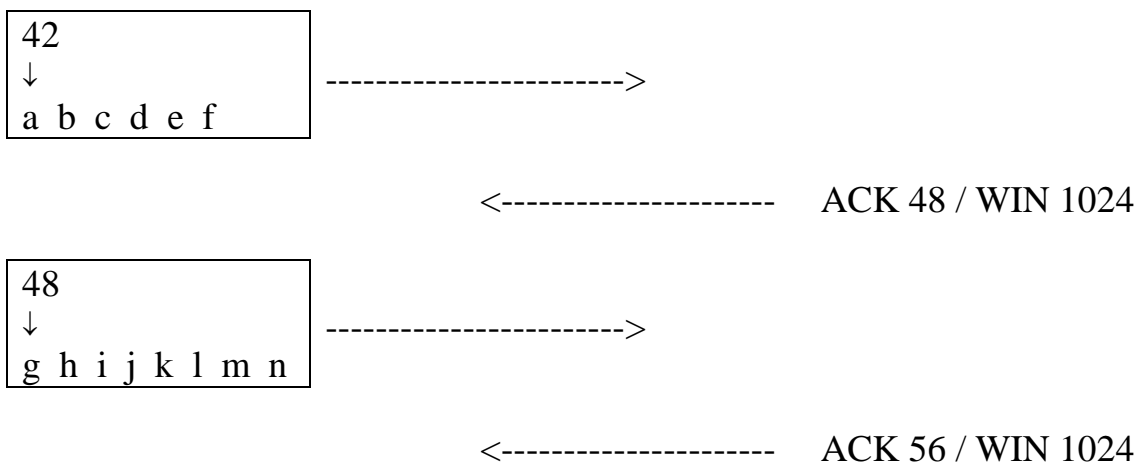
ACK 48	ACK 56
--------	--------

Controle de Fluxo no TCP

Objetivo: Tornar a velocidade de transmissão da máquina fonte compatível com a velocidade de processamento da máquina destino.

- Destinatário não pode simplesmente "segurar" o seu ACK. Isto causaria uma retransmissão.
- A saída é o destinatário definir um "tamanho de janela" disponível.

Exemplo: Caso o destinatário esteja recebendo normalmente, ele mantém o "*window size*"





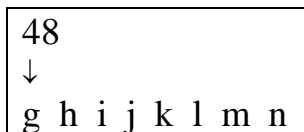
<----- ACK 62/ WIN 1024

- Alternativamente, se a estação destino pára de aceitar dados, os dados que chegam vão consumindo espaço do buffer e a WIN decresce.

<----- ACK 42/ WIN 1024



<----- ACK 48 / WIN 1018



<----- ACK 56 / WIN 1010

•
•
•

<----- ACK 1066 / WIN 0

- Fonte deve checar (a cada 2 mim) enviando um pacote de teste.

• Transferência Normal de Dados

Na máquina Fonte:

- Aplicação chama "*send (string)*"
- TCP envia pacote, a menos que a *WINDOW* do destinatário esteja fechada
- TCP guarda cópia do dado para possível retransmissão

Na máquina Destino:

- Pacote chega

- *Sum check* OK ? Se não, descarta
- O comprimento dos dados está dentro da janela ? Se não, descarta.
- Bufferiza dados para serem lidos pela aplicação
- Envia *acknowledgement*
- Quando o buffer é liberado, abre janela

TCP contém campos para:

- Flags (URG,ACK,PSH,RST,SYN,FIN)
- *Checksum* (detecção de erros)
- *Options* (usados para abertura de conexão)
- Data

Alguns FLAGS

- URG - Sinaliza ALGO que deve ser feito AGORA !
- RST - um erro ocorreu. Reinicialize a conexão
- FIN - Acabaram-se os dados, feche a conexão

Estabelecimento de Conexões - TCP

- Normalmente uma extremidade é passiva (um serviço esperando ser chamado- por exemplo, TELNET SERVER)
- Outra extremidade é ativa (um usuário começando uma sessão TELNET)
- É necessário comunicar Initial Sequence Number
- Estações geram ISN a cada conexão
- Utiliza TCP option para comunicar máximo tamanho do segmento

O Protocolo UDP - User Datagram Protocol

Útil para aplicação onde não se quer overhead no estabelecimento da conexão.
O Overhead também é minimizado pela simplicidade do header.

Dados são contidos num único pacote para transmissão
Sem aviso de recebimento; sem garantias
Pacote pode chegar danificado
Pacote pode não chegar
Pacote pode ser duplicado
Pode chegar fora da sequência

Apesar da aparente vulnerabilidade, o UDP é utilizado em importantes aplicações:

Routing na Internet

Name Service

É usado pesadamente no NFS da SUN

Source	Destination
Port	Port
Length	Checksum
Data	

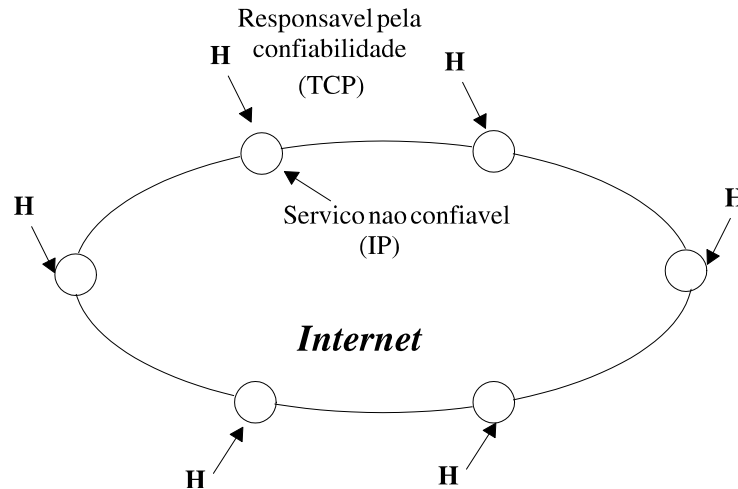
O UDP fornece três serviços:

Multiplexação

Length checking

Sumchecking

Roteamento de Datagramas IP



Routing: pode ser uma tarefa difícil, especialmente em máquinas com múltiplas conexões

O Routing Software escolhe o melhor caminho levando em consideração:

a carga na rede

o comprimento do datagrama

o tipo de serviço especificado no header

A maioria do software de roteamento é muito menos sofisticado e escolhe rotas baseado em considerações fixas sobre o caminho mínimo.

Routing é o processo de escolha de um caminho, enquanto que o **router** é o computador que executa o processo

Roteamento Direto

Transmissão de um datagrama de uma máquina diretamente para outra.

Roteamento Indireto

Ocorre quando o destino não está na mesma rede que a fonte.

Perguntas:

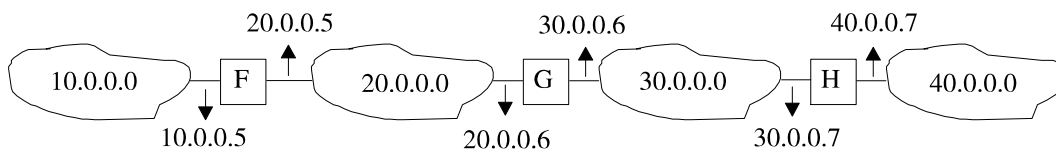
Como um gateway sabe para onde enviar um datagrama?

Como um host sabe qual gateway usar para um dado destino?

Tabelas de Roteamento

A tarefa de roteamento na Internet é simplificada pelo esquema de endereçamento que define **endereço de rede** e **endereço de host**.

Tipicamente uma tabela de roteamento contém pares (N,G), onde N é o endereço da rede de destino e G é o endereço de um gateway por onde devem ser enviados datagramas para a rede N.



Para o gateway G	
Alcança hosts na rede	Deve usar o endereço
20.0.0.0	entrega direta
30.0.0.0	entrega direta
10.0.0.0	20.0.0.5
40.0.0.0	30.0.0.7

Para manter as tabelas pequenas, o algoritmo de roteamento baseia suas decisões em endereços da **rede de destino** e não em **host de destino**.

Alguns pontos a considerar:

1. Todo tráfego para uma mesma rede segue o mesmo caminho (mesmo que existam caminhos alternativos)
2. Apenas o gateway final sabe se o host destino existe ou se é operacional
3. Porque cada gateway traça sua rota independentemente, deve-se garantir que comunicação em dois sentidos seja sempre possível

Rotas Default

Se uma rota não se encontra na tabela, segue uma rota padrão (default).

Rotas Específicas para Hosts

O TCP/IP permite também a definição de uma rota específica para determinado host.

Algoritmo usado no roteamento de um pacote IP (datagrama chegando)

1. Extraia o endereço Internet de destino (Id) do datagrama;
2. Extraia o endereço da Rede (In);
3. Se In é igual a qualquer endereço de rede diretamente conectada, envie o datagrama para aquela rede (mapeamento Id em Ifísico, encapsulamento e envio);
4. Senão, se Id aparece na tabela de rotas específicas, envie como mostrado na tabela
5. Senão, se In aparece na tabela de roteamento envie como mostrado;
6. Senão, se uma rota default existe, envie para o gateway adequado;
7. Senão, gere um erro de roteamento.

Lab1 - Configuração e Instalação de uma Subnet de Comunicação

15/09/95

1.0 Introdução

Nesta parte do curso, vamos fazer uma instalação de uma subnet de comunicação ethernet sobre cabo coaxial fino. A instalação de um sistema como este varia bastante dependendo do MAC utilizado. Mesmo em relação à ethernet, pode haver uma grande variação dependendo do meio utilizado (Twisted Pair, Thin coax, Thick coax, Fiber). Adicionalmente, placas de fabricantes diferentes podem ter características completamente diferentes (memória RAM mapeada ou I/O, configuração por straps ou por software, etc). Uma boa lida no manual de instalação é essencial para entender estas particularidades.

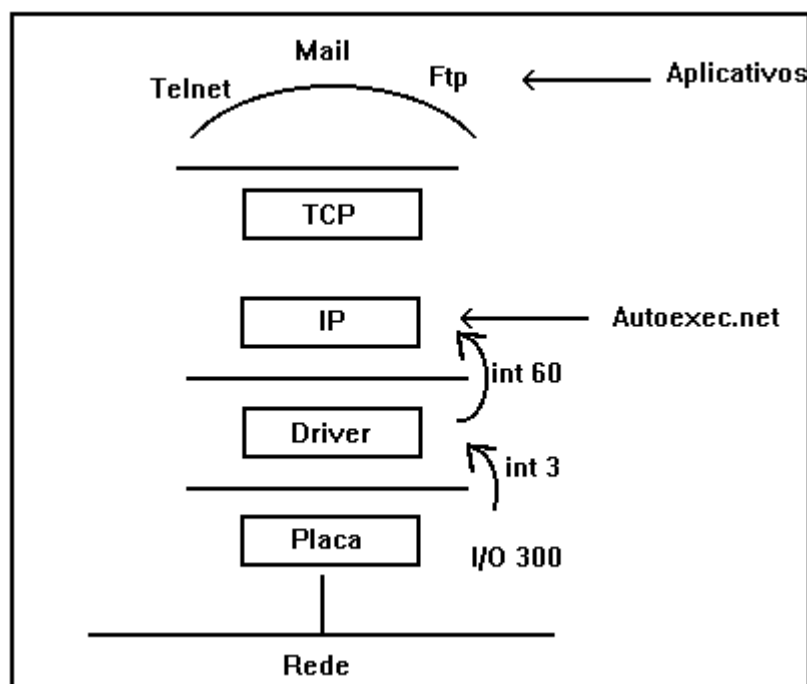
A subnet não faz nada sozinha. Ela apenas oferece os serviços de transporte de dados. Para que possamos fazer um teste efetivo do sistema teremos que colocar um driver e um protocolo de comunicação que ofereça alguns serviços extras, de forma que um aplicativo possa rodar de maneira confiável. O protocolo escolhido para o teste foi o TCP/IP. Existem várias implementações deste protocolo, para várias (quase todas) plataformas. E' um protocolo aberto, na medida que sua especificação está disponível para quem quiser implementá-lo; e, mais importante: sua especificação NÃO menciona qualquer característica que possa particularizá-lo para uma determinada plataforma ou sistema operacional.

A implementação utilizada nesta experiência é a conhecida como KA9Q. Foi escrita para ambiente DOS, mas teve como base a implementação feita para o 4.3 BSD UNIX. Uma de suas características principais é o seu núcleo que suporta múltiplos processos, de modo que várias conexões podem estar ativas concomitantemente. Apresenta também uma SHELL que serve de acesso aos usuários. É da shell que se dispara os comandos (TELNET, FTP, PING, etc). Também existem vários comandos para configuração e debugging.

Como dito anteriormente, as interfaces entre camadas são extremamente importantes em redes. Algumas delas, no entanto, assumem papel primordial:

- *A interface entre a placa e o driver* que define como o software vai "conversar" com o hardware. Deve haver uma concordância entre as duas partes em relação à sinalização de pedido de serviço e os pontos de acesso. Por exemplo, normalmente a placa sinaliza para a CPU a existência de um novo pacote através de uma interrupção (no exemplo abaixo, int3). No evento de uma int3, a CPU deve então rodar o driver para que ele faça a recepção do pacote. No exemplo, esta recepção é feita através de leituras de I/O na porta 300. Se a placa tivesse memória mapeada no espaço da CPU, tanto o driver quanto a placa deveriam concordar sobre o endereço desta memória.

- **A interface entre o driver e o protocolo principal (TCP/IP).** Estas duas entidades são, normalmente, carregadas em separado e, portanto, também devem ser configuradas de comum acordo. Como agora se quer comunicar entre dois programas, necessitaremos utilizar uma interrupção de software. O driver utilizado (packet driver) aceita qualquer endereço no espaço entre 60 e 80. Se duas placas são utilizadas (como num roteador), 2 drivers devem ser carregados e duas interrupções para hardware e duas para software devem ser usadas.
- **A interface entre o protocolo principal e os aplicativos.** A maioria das implementações de TCP/IP fornecem interfaces de programação, as chamadas API (Application Program Interface). A API mais conhecida para TCP/IP são os sockets. UNIX, KA9Q e Windows disponibilizam sockets para que usuários possam escrever programas. Esta interface não será explorada neste curso.



1.1 Cuidados Iniciais

- * **Leia o manual da placa;**
- * **Veja quais dados vão precisar ser configurados e se a placa precisa configuração manual ou por software;**

A placa ethernet será conectada ao duto do micro. Antes de instalá-la (na verdade, esta regra deve ser observada para qualquer dispositivo de I/O instalado no micro), deve-se observar quais interrupções, quais endereços de I/O e quais porções de memória estão livres e, portanto, podem ser alocadas para a nova placa. Existem várias ferramentas que mostram o mapa do sistema: ckeckit, norton, etc. *Use uma destas ferramentas para decidir como configurar a placa.* Note, por exemplo, que outros dispositivos já estão utilizando interrupções e endereços de I/O. COM1, controladora de disco e de vídeo são exemplos.

1.2 Identificação dos Componentes na Placa

* Verifique (com a ajuda de um monitor os diversos componentes da placa. Em especial, verifique o controlador, o sistema de interfaceamento com o duto da CPU e a interface com o meio físico.

1.3 Configuração e instalação da Placa

- * Configure a placa escolhendo: interrupção, endereço de I/O, e endereço de memória, se houver.
- * Certifique-se que o micro esteja desligado;
- * Instale a placa no duto. Se a placa for configurável apenas por software, siga as instruções para configurá-la.
- * Religue o micro.

2.4 Configuração do Software

O próximo passo é a instalação do driver e do KA9Q. Você deve ter anotado os parâmetros de configuração da placa, para informar ao driver. Supondo que o programa driver seja o ne2000, o seguinte comando deve ser teclado (para a configuração do exemplo):

ne2000 0x60 0x3 0x300

Crie um diretório ka9q (poderia ser qualquer nome!) e coloque o KA9Q. Ele é composto por alguns arquivos. Os principais são: net.exe (é o núcleo do sistema); autoexec.net (é o arquivo de configuração); ftpusers (é o arquivo onde se colocam os usuários, seus passwords e suas permissões). Apenas usuários registrados em ftpusers serão atendidos remotamente.

O arquivo **autoexec.net** deve conter algo do tipo:

<i>hostname</i>	(escolha um nome para a sua máquina)
<i>ip address [200.100.100.NN]</i>	Troque NN por um número entre 0 e 64
<i>attach packet 0x60 le0 8 1500</i> <i>e</i>	Nesta linha você informa o tipo de driver (packet) o número da interrupção de software
<i>ifconfig le0 ipaddress 200.100.100.NN broadcast 200.100.100.00 netmask 0xffffffffc0</i>	
<i>route add [200.100.100.00]/26 le0</i> <i>route</i>	Este comando diz como os pacotes serão roteados Quando iniciar, o net vai mostrar as rotas
<i>smtp timer 1200</i>	
<i>tcp mss 216</i>	
<i>log net.log</i>	
<i>tcp window 432</i>	
<i>start telnet</i>	
<i>start ftp</i>	
<i>start echo</i>	

1.5 Teste de funcionamento

Por default, o **net.exe** vai procurar o **autoexec.net** no diretório raiz. Para fazê-lo ler do diretório **ka9q**, faça

```
net -d \ka9q
```

Para testar a rede, faça **PING** para a máquina vizinha (voce deve dar o número dela no comando). O sistema responde com o **rtt** (round trip time), tempo gasto entre ir na máquina remota e voltar (em milissegundos).

Uma vez funcionando, você pode fazer outros comandos como por exemplo, *ftp*. Certifique-se que voce está registrado no arquivo **ftpusers** da máquina remota.

Divirta-se!

Lab2 - Instalação, configuração e utilização de uma API TCP/IP em Windows

27/10/95

1.0 Introdução

Neste lab, usaremos uma implementação de sockets para Windows (TRUMPET) para rodar aplicações (TCP||UDP)/IP em ambiente Windows. WinQVT (que fornece um ambiente amplo para acesso à Internet: telnet,ftp,etc) sera' utilizado como exemplo. Numa aula futura será instalado o Netscape para acesso ao WWW.

No último lab, a implementação TCP/IP utilizada foi KA9Q, para ambiente DOS. Aquela implementação era completa: o corpo do protocolo e os aplicativos faziam parte do mesmo pacote. No Windows (Argh!!!), as coisas são um pouco diferentes e existem várias alternativas para a obtenção do mesmo resultado. Nós vamos utilizar a mais complicada!!! Ao invéz de utilizarmos um packet driver (que foi originalmente desenvolvido para rodar TCP/IP em micros sem windows) como na última experiência, vamos utilizar o driver original feito pela NOVELL so' para rodar IPX e NETX. Dai então teremoss que rodar algumas outras peças de ssoftware para prover as capacidades esperadas do driver.

Note que muito da complicação da instalação e' devida ao fato de se estar aproveitando programas que foram desenvolvidos para um fim numa aplicação diferente. As camadas de baixo vão ter que ter capacidades para gerenciar o acesso DOS, Windows e TCP/IP.

Esta experiência então, consiste em:

- instalar e testar um cliente NOVELL para DOS;
- testar a conectividade do servidor dentro do ambiente Windows;
- configurar, instalar e testar uma implementação (TRUMPET) de TCP/IP;
- instalar um aplicativo (WinQVT) que utilize a API (socket) fornecida pelo TRUMPET.

2.0 As camadas

- **Driver Nativo**, define a maneira como o software conversará com o hardware, deve ser set'ado tanto na placa (se necessário) como nos arquivos de configuração para o driver da placa (net.cfg). Os valores para endereço de porta e interrupção de hardware devem ser providos. (**Arquivo NE2000.com**)

- **Driver “Universal”**, o programa NE2000.com foi desenvolvido especificamente para um cliente NOVELL numa máquina DOS. Como vamos querer que outros protocolos utilizem o mesmo driver, temos que fazer algumas adaptações. O arquivo **lsl.exe** prove esta capacidade ao driver

- **A pilha Novell**

A pilha NOVELL é completada com o protocolo propriamente dito (IPXODI.com) e com a shell de acesso (NETX.com).

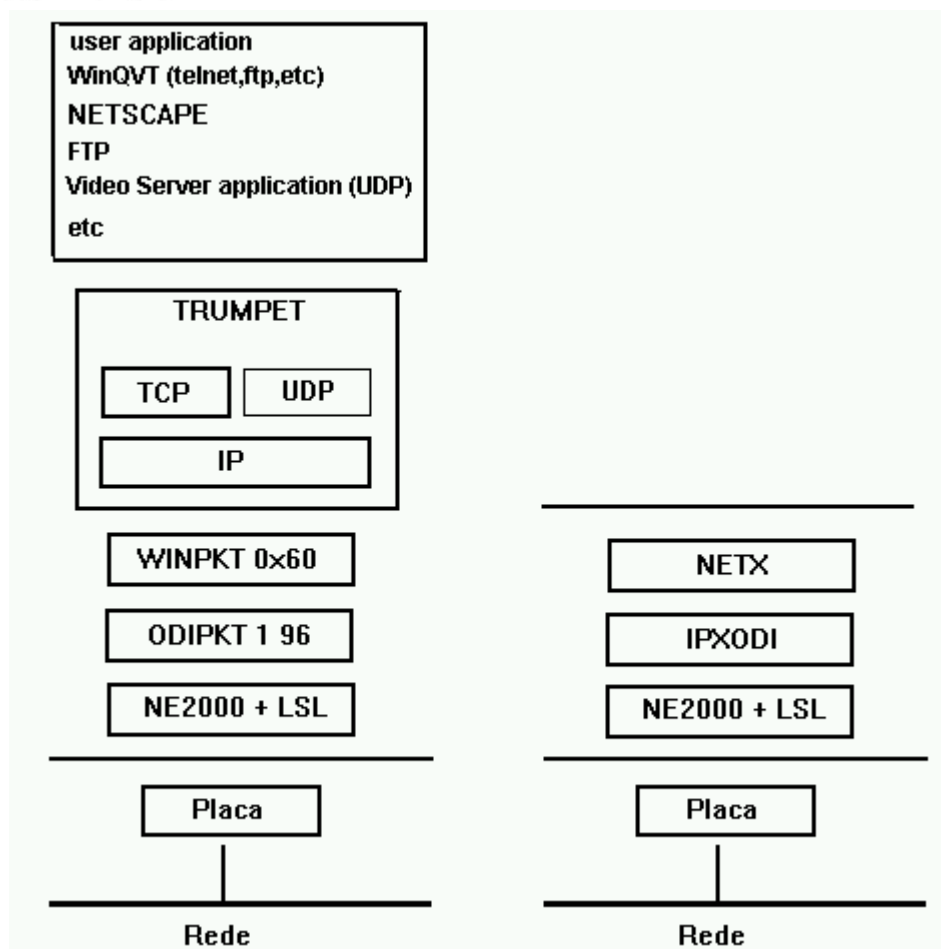
- **Packet Driver**. Como vamos utilizar TCP/IP também, devemos acrescentar as capacidades de Packet Driver ao driver “nativo/universal”. Isto é feito pelo **ODIPKT.com**

- **A pilha TCP/IP**

Para colocar a pilha TCP/IP no Windows, teremos que acrescentar ainda mais um pedaço de software no driver “nativo/universal/packet”. É o **winpkt.com**. Agora podemos colocar o TCP/IP propriamente dito (**TRUMPET**)

- **Os aplicativos**

Agora, aplicativos usando winsock podem rodar nesta máquina. O WinQVT é um deles.



3.0 Instalação da Placa

(((vide apostila passada))))

4.0 Preparação de arquivos para os clientes Novell e TCP/IP em Windows

Deve ser criado um diretório **REDES95**. Devemos então copiar o driver nativo (NE2000.com, NE1000.com ou 200E.com) dependendo da placa de rede utilizada. Além disso o driver universal (LSL.com), packet driver (ODIPKT.com) e finalmente o WinPKT.com

5.0 Instalação de um cliente Novell

Após configurada a placa corretamente, deve ser editado um arquivo de configuração para o driver nativo, é o NET.CFG. Nele, deve ter os valores para interrupção de hardware e porta de comunicação dependendo de como foi configurada a placa.

```
Link Suport
  Buffers 8 1500
  MemPool 4096
```

Link driver NE2000

```
int #1 5
port #1 320
Frame ETHERNET_802.3
Frame ETHERNET_II
```

Agora podemos carregar na ordem os drivers nativo, universal, packet driver e a shell de acesso. Exemplo :

```
LSL                                // driver universal
NE2000                             // driver nativo, olhará no net.cfg
ODIPKT 1 96                        // 96 == 0x60
NETX                               // shell de acesso
```

Uma vez em funcionamento, um drive novo aparecerá (o primeiro disponível) permitindo o acesso a um servidor. Podemos então logar entrando nesse drive (por exemplo F:) e utilizando o comando login.

É interessante, uma vez logado no servidor, a verificação dos recursos disponíveis até em Windows. Podemos verificar isso pelo Gerenciador de Arquivos.

5.0 Upgrade para um cliente TCP/IP

Para rodar a pilha TCP/IP, precisaremos de um driver adicional, o WinPKT com o parâmetro da interrupção de software estipulada no ODIPKT.

```
ODIPKT 1 96                        // (decimal:)
WINPKT 0x60                       // (hexa:)
```

Agora podemos rodar o Windows, executarmos e configurarmos o Trumpet (dentro do menu setup)

IP da máquina	143.107.231.(150 - 158)
Gateway	143.107.231.129
Name Server	143.107.231.1 (xavante)
máscara	255.255.255.224

Packet Vector 60

Finalmente podemos instalar uma aplicação, o WinQVT que utiliza da API Windows Sockets fornecida pelo Trumpet.

Lab3 - Roteiro para Instalação de um Servidor NOVELL

24/11/95

1. Objetivos:

Os objetivos dessa aula prática é mostrar aos alunos os passos necessários para a instalação de um servidor de rede utilizando o sistema operacional da Novell, o Netware. Também será demonstrado como fazer a instalação das estações de trabalho, inclusive com a opção de instalação de estações com boot remoto, e instalar o servidor como sendo um roteador IP.

2. Primeiros Passos: A escolha do hardware para o servidor

As versões do Netware 3.x e superiores exigem um servidor de arquivos dedicado, isto é, o servidor não poderá ser utilizado como estação de trabalho. O hardware mínimo requerido para essas versões é:

- microcomputador 386 ou superior;
- 4 Mb de memória RAM (recomendado 8 Mb);
- 80 Mb de espaço em disco rígido;
- monitor padrão CGA (VGA recomendado);
- Placa de rede compatível com o padrão NE2000;
- MS-DOS 3.x ou superior.

e para as estações de trabalho é o seguinte:

- microcomputador PC/XT ou AT x86 (máquinas 386 ou superiores são recomendadas);
- 640 Kb de memória RAM (recomendado 4 Mb);
- Uma unidade de disco flexível;
- monitor padrão CGA (VGA recomendado);
- Placa de rede compatível com o padrão NE2000;
- MS-DOS 3.x ou superior.

É bom lembrar que todo o processamento será feito nas estações de trabalho, portanto, dependendo das tarefas a serem executadas uma estação bem configurada é sempre importante.

3. Instalação do Servidor

Para instalação do servidor de arquivos deve-se seguir os seguintes passos:

1. Criar uma partição DOS no disco rígido e formatá-la com sistema operacional desejado:
 - Inicialize o computador com um disco contendo o sistema operacional e no mínimo os utilitários FDISK e FORMAT.
 - execute o utilitário FDISK para criar a partição DOS no disco rígido. A partição DOS deve ter no mínimo 1,5 Mb e no máximo 3 Mb de espaço. Não se esqueça de ativar a partição.
 - Formate a partição criada com a opção /s para gerar o sistema operacional no disco rígido.
2. Copie o conteúdo do disco 1 de instalação do software para o disco rígido
3. Execute o programa SERVER
 - entre com o nome do servidor (de 2 a 47 caracteres incluindo qualquer caracter alfanumérico, hífen e underscores. Não pode conter ponto e espaços em branco.
 - entre com o número de rede interno (Internal Network Number). Este número deve ser único na rede e no formato hexadecimal e deve ter de 1 a oito dígitos. Exemplo: AAAF.
4. Execute os módulos NLM necessários
 - Primeiro carregue o driver de disco que você esta utilizando. Execute o comando:
LOAD [path] disk_driver <Enter>
onde o path é o drive e o caminho completo de onde está localizado o driver da placa de rede, e disk_driver é o driver para o disco que está instalado.
5. Execute o programa INSTALL
 - LOAD INSTALL
6. Crie as partições Netware desejadas.
 - Para isso entre na opção “Disk Options” dentro do menu “Installation Options”
 - Escolha a opção “Partition Tables”
 - Escolha a opção “Create Netware Partition”. O Netware permite apenas uma partição netware por disco e aloca automaticamente 98% do espaço definido como espaço disponível e 2 % como Hot Fix Redirection Area

7. Crie os volumes

- Escolha a opção “Volume Options” no menu “Installation Options”
- Digite <Insert> na janela “Volumes”. O Netware cria automaticamente o volume SYS como sendo o primeiro volume.
- Na janela “New Volume Information” escolha as características do volume que você está criando. O Netware reserva por default todo o espaço disponível no disco para cada partição que você estiver criando.
- Tamanho do Bloco (Volume Block Size):

O bloco é a unidade de armazenamento de dados. O tamanho padrão para o bloco é 4 Kb (4096 bytes). Blocos pequenos requerem uma maior quantidade de memória no servidor para gerenciar a FAT e a tabela de diretórios; mas se você tem arquivos pequenos menos espaço em disco será perdido com os blocos não preenchidos. Blocos grandes são úteis quando se trabalha com grandes arquivos de dados.

- Determine o tamanho do espaço ocupado por cada volume.
Deve-se planejar o tamanho de cada volume dependendo das aplicações a serem desenvolvidas e do espaço requerido para cada usuário. O cálculo do espaço dos volumes é dado pela seguinte fórmula:

$$\text{espaço do volume} = (1024 / \text{tamanho do bloco}) \times \text{quantidade de espaço desejada}$$

- Digite <escape> e responda “YES” para cada volume.

8. Copie os arquivos Públicos e de Sistema

- Retorne ao menu “Installation Options”
- Selecione a opção “System Options” no menu “Available System Options”
- Insira os discos solicitados até o fim da instalação

9. Carregue os drivers de rede

- Saia do programa de instalação e no prompt digite:

```
load driver int=int_number port=i/o_port_number
```

driver = nome do driver da placa de rede

int_number = interrupção da placa de rede

port_number = I/O address da placa de rede

- execute o programa bind para associar os pacotes IPX para cada placa de rede

```
bind ipx to lan_driver
```


10. Crie os arquivos AUTOEXEC.NCF e STARTUP.NCF

- O arquivo AUTOEXEC.NCF é similar ao autoexec.bat do DOS. Ele executa todas as instruções necessárias para a instalação correta dos programas do Netware. Como exemplo de como este arquivo funciona mostramos o AUTOEXEC.NCF do servidor Ensino do ICMSC.

```
file server name ENSINO

ipx internal net 1

load NE1000 int=2 port=340 name=ENSPESQIPX
bind IPX to ENSPESQIPX net=3

load NE1000 int=3 port=300 name=POSGRAD
bind IPX to POSGRAD net=64

load 386ME-16 int=4 port=360 name=GRAD
bind IPX to GRAD net=128

load PN-16CT int=Ch port=2C0 name=GRAD_1
bind IPX to GRAD_1 net=160

load NE1000 int=2 port=340 frame=ETHERNET_II name=ENSPESQ
load NE1000 int=3 port=300 frame=ETHERNET_II name=NOVELL_POSG
load 386ME-16 int=4 port=360 frame=ETHERNET_II name=NOVELL_GRAD
load PN-16CT int=5 port=280 frame=ETHERNET_II name=NOVELL_POS1
load PN-16CT int=Ch port=2C0 frame=ETHERNET_II name=NOVELL_GRAD1

load TCPIP forward=YES

bind IP to ENSPESQ      addr=143.107.231.7   mask=ff.ff.ff.e0
gate=143.107.231.13
bind IP to NOVELL_POSG  addr=143.107.231.65   mask=ff.ff.ff.e0
bind IP to NOVELL_POS1  addr=143.107.231.97   mask=ff.ff.ff.e0
bind IP to NOVELL_GRAD  addr=143.107.231.129  mask=ff.ff.ff.e0
bind IP to NOVELL_GRAD1 addr=143.107.231.161  mask=ff.ff.ff.e0

load REMOTE teste
load RSPX
mount all
#load PSERVER TEC
6.load mercury
load mercurys
load mercuryc
load tcpcon
#disable login
```

- O arquivo STARTUP.NCF é carregado sempre antes do AUTOEXEC.NCF e contém comandos para carregar o driver do disco rígido. Ele fica gravado no drive C: e é carregado logo após o arquivo SERVER.EXE.

```
load AHA1540 port=330 int=B dma=5
```

4. Instalação das estações de trabalho

1. Verifique se a sua estação de trabalho possui uma placa de rede compatível com o padrão Novell e qual o sistema operacional que está sendo executado.
2. Cheque os valores da interrupção e o endereço de I/O
3. Certifique-se de que possui os drivers adequados para esta placa.

- Cada fabricante de placas oferece junto com o produto todos os drivers necessários para conexão como servidor;
- Para facilitar podemos utilizar o ODI (Open Data-Link Interface)

4. Crie um diretório NET e copie os seguintes arquivos para ele:

- IPXODI.COM (fornecido pela Novell)
- LSL.COM (fornecido pela Novell)
- Driver da Placa de Rede (fornecido pelo fabricante da placa)
- NETX.COM (fornecido pela Novell)

5. Crie um arquivo chamado NET.CFG com os seguintes comandos:

```
Link Suport
    Buffers 8 1500
    MemPool 4096

Link driver NE2000
    int #1 5
    port #1 320
    Frame ETHERNET_802.3
    Frame ETHERNET_II
SHOW DOTS = ON
```

6. Execute os seguintes comandos, na ordem em que aparecem

```
lsl
ne2000
ipxodi
netx
```

7. Mude para o drive F: e execute o comando LOGIN

```
login supervisor
```

8. Agora você está pronto para administrar a sua rede utilizando os diversos comandos de gerenciamento disponíveis no Netware.