Apostila da disciplina: "Fundamentos e Infra-estrutura em Redes de Computadores"

1. Introdução:

Segundo a Ementa da disciplina temos os seguintes itens a ser apresentados nesse período de aula: "Classificação de Redes de Computadores (MAN, WAN, LAN). Modelo de Referência OSI da ISO. Arquitetura de Redes (TCP/IP e proprietárias). Redes públicas de comunicação de dados (tipos, padrões, utilização). Interligação de redes. Protocolos. Projeto de Redes".
Busca-se seguir a seguinte programação:

1. Teórica de Base (1.1. *Hardware* de Rede, 1.2. *Software* de Rede, 1.3. Classificação de Redes e Arquitetura, 1.4. Teoria de Telecomunicações).
2. Teórica de Modelos de Referência (2.1. Modelos ISO/OSI, TCP/IP, 2.2. Camadas Física e Enlace, 2.3. Camadas Redes e Transporte, 2.4. Camada de Aplicação).
3. Prática de Redes (3.1. Entendimento de pacotes, 3.2. Tráfego de redes e diretrizes, 3.3. Suporte ao roteamento de pacotes, 3.4. Diretrizes de cabeamento estruturado)
4. Segurança da Informação (4.1. Problemas de Segurança na Comunicação,4.2. Criptografia, 4.3. Criptografia Simétrica, 4.4. Chaves Públicas e Gerenciamento de Chaves

Parte 1: Fundamentos, Histórico, Classificações e Motivações

1. Parte Histórica

Cada um dos 3 séculos anteriores foi dominado por uma única nova tecnologia:

- **Século XVIII**: Foi a época dos grandes sistemas mecânicos que acompanharam a Revolução Industrial.
- **Século XIX**: Máquinas a Vapor, consolidação da industrialização padronizada de massa.
- **Século XX:** As principais conquistas tecnológicas se deram no campo da aquisição, do processamento e da distribuição de informações. Instalação de redes de telefonia em escala mundial, a invenção do rádio e da TV, o nascimento e o crescimento da informática.
- **Século XXI:** Evolução de modo transparente das tecnologias convergindo rapidamente. Diferenças de coletas, armazenamento e transporte estão desaparecendo rapidamente, Organizações com escritórios geograficamente distribuídos rapidamente se comunicam, conversão das tecnologias ao modelo ETHERNET.

Apesar da indústria de informática ser jovem em comparação a outros setores (automobilístico e aéreo), foi simplesmente espetacular o progresso que os computadores experimentaram em curto período. Durante as primeiras décadas de sua existência, os sistemas eram altamente centralizados em geral instalados em uma grande sala, muitas vezes com acesso restrito.

A fusão dos computadores e comunicações teve uma profunda influência na forma como os sistemas computacionais são organizados. O conceito então dominante de

"centro de computação" como uma sala com um grande computador aos quais os usuários levam seu trabalho para processamento agora está totalmente obsoleto. Este conceito foi alterado para em vez de um computador processando tudo centralizadamente para que trabalhos são realizados por um grande n° de computadores separados, porém interconectados. Esses sistemas são chamados de **redes de computadores.**

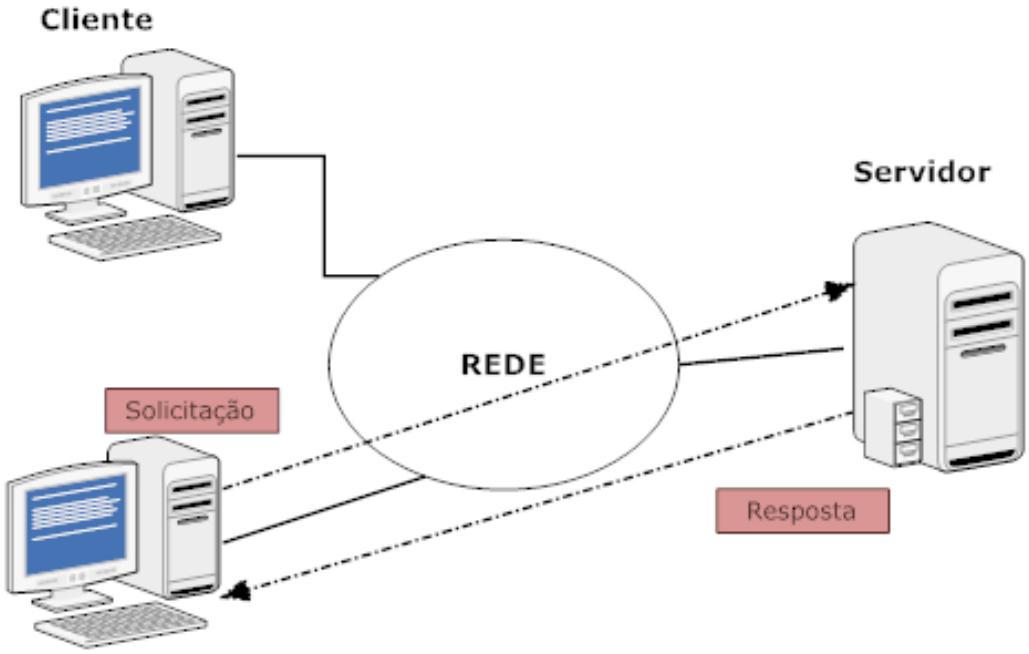
Redes de computadores envolvem equipamentos, protocolos, modelos de referência para uso em hardwares de rede, já os **sistemas distribuídos** é responsável pela implementação de um modelo coerente em *software*, em outras palavras, é um sistema de *software* instalado na parte superior de uma rede dando alto grau de coesão e transparência. Exemplo: World Wide Web sob ETHERNET. Em um sistema distribuído a existência de vários computadores autônomos é transparente para o usuário ele entra com um comando e é o SO que decide qual processador ou PC executará, enquanto que em redes o usuário que decide em qual PC fará o acesso para execução.

A ARPANET: A história começa no final da década de 1950. No auge da Guerra Fria, o DoD dos EUA queria uma rede de controle e comando capaz de sobreviver a uma guerra nuclear. Nessa época todas as comunicações militares passavam pela rede de telefonia pública, considerada vulnerável. Na década de 1960, o DoD firmou um contrato com a RAND Corporation para encontrar uma solução. Um de seus funcionários Paul Baran, apresentou um projeto altamente distribuído e tolerante a falhas, tendo em vista que os caminhos entre duas centrais de comutação quaisquer poderia ser feito por mais de uma rota. A idéia foi meio que ignorada por parceiras como AT&T (líder telefonia). Porém os americanos verificaram que estavam atrás na corrida espacial quando os soviéticos lançaram o primeiro satélite artificial denominado Sputnik em 1957. Desse modo os EUA criaram no pentágono uma divisão denominada ARPA (*Advanced Research Projects Agency*). Tal agência realizava seu trabalho oferecendo concessões a empresas e universidades para desenvolvimento conjunto para o DoD. E foi assim que verificaram que seria interessante assim como em outros países já existiam iniciativas de criar uma rede digital de comutação de pacotes. Desse modo em 1967 foi criada a ARPANET. Inicialmente em 56 Kbps e em 1972 já interconectava costa a costa americana. No final dos anos 70, a ARPANet tinha crescido tanto que o seu protocolo de comutação de pacotes original, chamado *Network Control Protocol* (NCP), tornou-se inadequado. Foi então que a ARPANet começou a usar um novo protocolo chamado TCP/IP (*Transmission Control Protocol/Internet Protocol*).
ARPANet divide-se e origina a MILNET -- para assuntos militares -- e o restante da rede torna-se pública e tem seu nome alterado para Internet.

2. Usos de Redes de Computadores

2.1 Aplicações Comerciais: Foco no compartilhamento de recursos, o objetivo é deixar programas, equipamentos e especialmente dados ao alcance de seus usuários. Ex: Compartilhamento de impressora e servidor de arquivos.

Modelo Cliente/Servidor



Exemplos de redes comerciais: e-mail, Remote Desktop, VoIP, e-commerce.

Formas de e-commerce:

Abreviação	Significado	Exemplo
B2C	Business to Consumer	Compra *online* (livros, DVDs)
B2B	Business to business	Fábrica solicitando pneu a fornecedor
G2C	Government to consumer	Governo distribuindo formulários impostos
C2C	Consumer to consumer	Mercado livre, OLX.
P2P	Peer-to-Peer	Compartilhamento arquivos

2.2 Aplicações Domésticas: Em 1977, Ken Olsen era presidente da *Ditial Equipment Corporation*, então o segundo maior fornecedor de computadores de todo o mundo (depois da IBM). Quando lhe perguntaram por que a Digital não estava seguindo a tendência de mercado de computadores pessoais ele respondeu: "Não há nenhuma razão para qualquer indivíduo ter um computador em casa". A história mostrou ao contrário e a Digital não existe mais. As pessoas compravam antigamente computadores para processamento de textos, planilhas, programas específicos e jogos. Atualmente a maior motivação é a conexão a Internet. Agora muitos dispositivos móveis tem acesso a grande rede sendo a era da **conectividade.**

Ex. de aplicações Domésticas: Acesso a WWW, peer-to-peer, Messengers, Redes Sociais, Wikis, Vídeos sob demanda, Computação Ubíqua, RFID, etc.

3. Questões Sociais:

- Redes sociais: compartilhamento de idéias com indivíduos com o mesmo pensamento, será? As vezes o que é inofensivo para a gente poderá ser ofensivo para outra pessoa, principalmente questões relacionadas a credo, raça, time de fuutebol.
- A criação de perfis em empresas para acesso restrito ou irrestrito de informações dos usuários. Um administrador pode bloquear um tipo de serviço a determinado tipo de usuários, mas o patrão tem o direito de ler o que o funcionário escreve

em um simples email somente por estar utilizando de dentro de sua empresa? Isso é invasão de privacidade?

- Roubo de identidade na *web*: Pessoas ou computadores podem se passar por outros indivíduos na realização de tarefas automáticas. Por isso uso de CAPTCHAs.

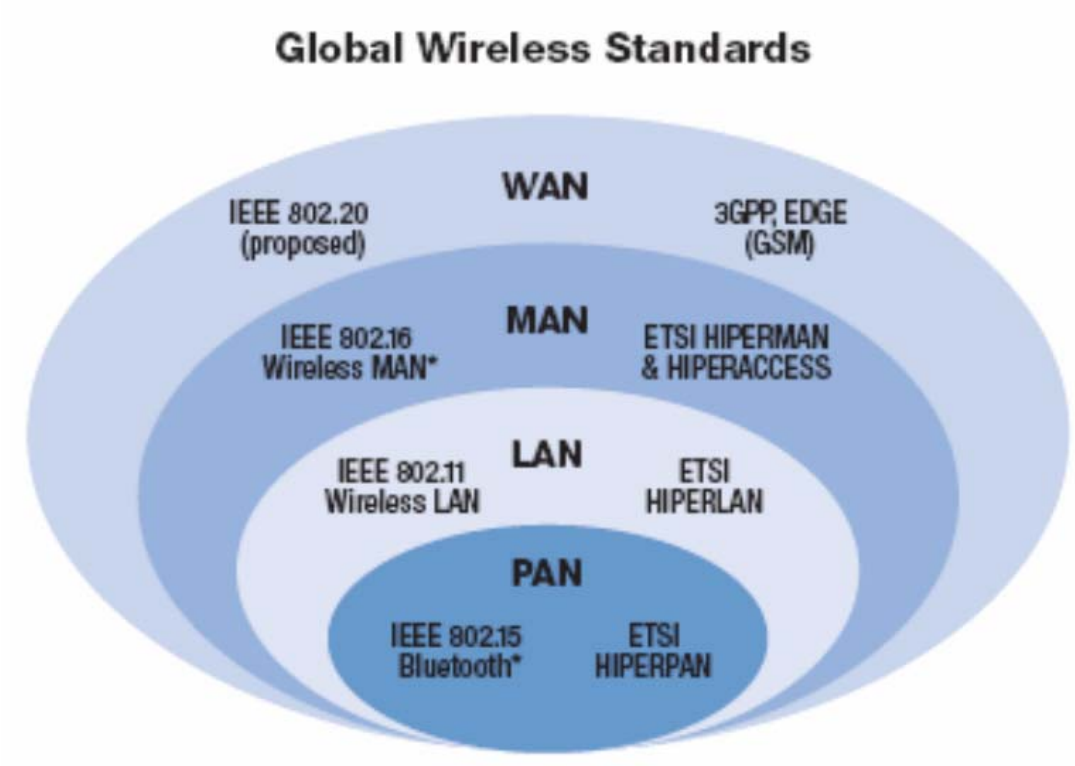
4. Hardware de Rede:

Não existem taxonomia para classificação de redes.
Classificações de redes quanto a **TECNOLOGIA DE TRANSMISSÃO E ABRANGÊNCIA**:

4.1 Tecnologias de Transmissão:

- UNICASTING: Ponto a ponto conectam pares de máquinas individuais. Envio de um transmissor para um receptor. Para nós conectados não vizinhos são realizados *hops* (pulos).
- BROADCASTING (desenhar): Um canal de comunicação compartilhado por todas as máquinas, os pacotes podem ser enviados por qualquer máquina e são recebidos por todas as outras, porém senão foi endereçado a ela é descartado. Alguns sistemas permitem o envio para um grupo ou subconjunto de máquinas, essa modalidade é chamada de MULTICASTING.

4.2 Escala (Abrangência):



- **LAN:** Normalmente redes privadas, contidas em alguns KM de extensão. Velocidade de 10 a 100 MBps.
- **Ligação INTEREDES**: É realizada com equipamentos especiais denominados **GATEWAYS.** Esse GW realiza as conexões e faz as conversões necessárias de HW e SW.

Antigamente o hardware de rede é que tinha importância, porém a estruturação dos softwares de rede ficou deliberadamente estruturada, portanto faz-se necessário termos ciência de alguns termos que serão utilizados no decorrer do curso todo.

5. Software de Rede:

5.1 Hierarquia de Protocolos: Para redução da complexidade do projeto de rede, a maioria é organizada como uma pilha de camadas (ou níveis), colocadas uma sobre as outras. O n°, nome, funções diferem de uma para outra camada de maneira bem definida. O objetivo de cada camada é fornecer determinados serviços as camadas superiores.

Quando uma camada *n* se comunica com outra camada *n* de outra máquina, as regras e convenções dessa comunicação recebem o nome de **protocolo (conjunto de primitivas de serviços).** Basicamente, um protocolo é um acordo entre as partes para uma comunicação. Ex. Protocolo de aperto de mão, beijo, diferente de uma princesa para um buteco.

Entre cada camada existe uma **interface** que define as operações e serviços da camada inferior para a superior.

Um conjunto de camadas e protocolos é chamado de **arquitetura de rede.**

Ex. Comunicação entre duas pessoas que não falam uma língua em comum, pág. 19.

5.2 Confiabilidade

- Questões de projeto relacionadas às camadas: Relacionadas à **Confiabilidade**

- **Detecção de Erros:** Como termos um mecanismo de localização de erros? Podem ocorrer problemas físicos, elétricos, como garantir que um *bit* que foi enviado de uma máquina chegou corretamente ao destino?
- **Correção de Erros:** Se encontramos um erro na comunicação como corrigi-lo?
- **Roteamento:** Como podemos saber um caminho que funcione corretamente para o envio de uma MSG?
- **Endereçamento:** Como endereçar corretamente uma máquina de destino ou diversas máquinas de destino?
- **Escalabilidade:** Como ter um projeto de rede que seja possível crescer quando a cidade/país/empresa tiver um crescimento?
- **Alocação de Recursos:** como um transmissor rápido envia informações para um receptor lento? Há uma necessidade de **controle de fluxo.**
- **Congestionamento:** A rede pode ficar sobrecarregada quando muitos computadores querem enviar tráfego e a rede não supri tal demanda, Omo resolver isso?
- **Qualidade de Serviço:** Como manter a qualidade do serviço de aplicações que são importantes para a organização, existe como?
- **Confidencialidade:** Como manter a confidencialidade na comunicação entre *hosts* da mesma organização?

5.3 Primitivas de Serviços: Um serviço é especificado formalmente por um conjunto de primitivas (operações) disponíveis para que os processos do usuário acessem o serviço. Essas primitivas informam ao serviço que ele deve executar alguma ação ou relatar uma ação executada por uma entidade par. Essas primitivas podem ser usadas para uma interação de solicitação/resposta em uma ambiente cleinte/servidor. Mostrar um *handshake* (LISTEN, CONNECT, RECEIVE, SEND, DISCONNECT).

5.4 Tipos de Serviços: As camadas podem oferecer dois tipos de serviços as camadas superiores.

5.4.1 Serviços orientados a conexão: Como a chamada telefônica: para falar com alguém você tira o telefone do gancho, tecla o telefone de destino, fala e em seguida desliga. Da mesma forma para conexões como se fosse um tubo: o transmissor empurra objetos (bits) e o receptor recebe do outro lado. Na maioria dos casos a ordem é preservada, de forma que os bits chegam na sequência como saíram da origem.

5.4.2 Serviços não orientados a conexão: Como no sistema postal, cada mensagem (carta) carrega o endereço de destino completo e cada uma delas é roteada pelo nós intermediários do sistema, independentemente das outras. Essa "carta" no sistema não orientado a conexão recebe o nome de DATAGRAMA.

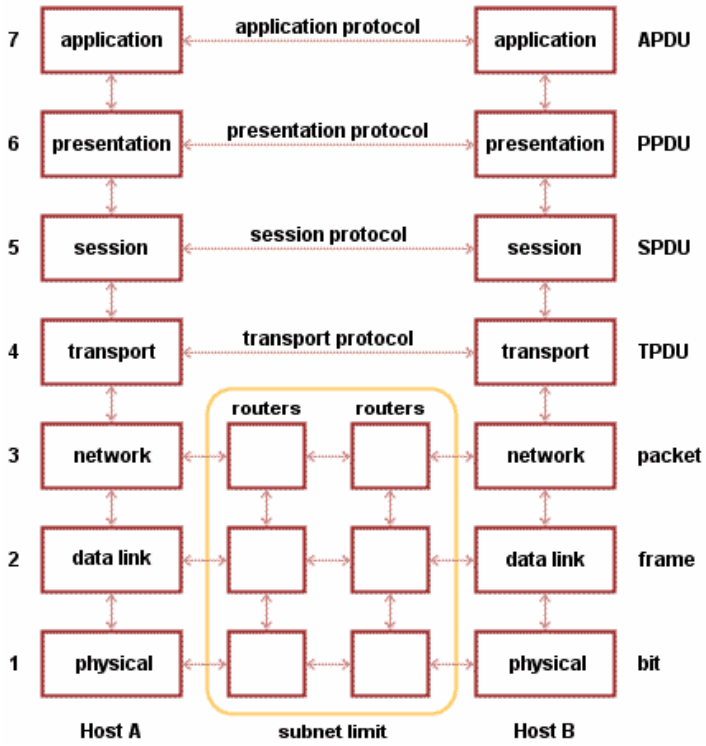
QP: Qual a principal diferença entre comunicação não orientada a conexão e orientada?

Parte 2: Modelos de Referência

1. Modelos de Referência:

Dois são os modelos a serem apresentados: OSI e TCP/IP.

O modelo ISO/OSI: Modelo de referência ISO OSI (*Open System Interconnection*), pois ele trata da interconexão de sistemas abertos, ou seja, sistemas abertos com outros sistemas. Modelo de 7 camadas (Camada Física, Enlace, Rede, Transporte, Sessão, Apresentação, Aplicação). Colocar Figura 1.17.

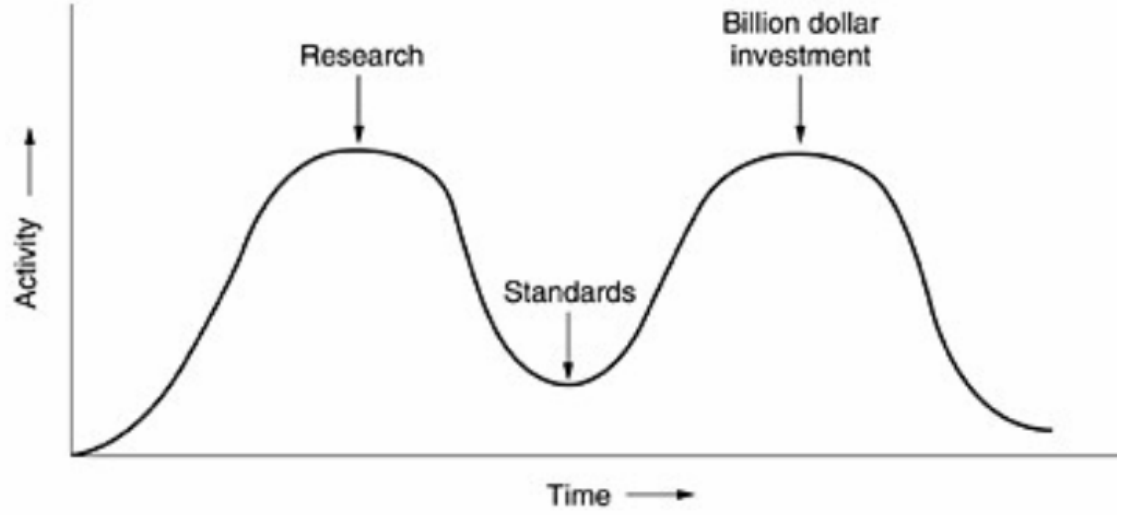


- **Camada Física:** Trata da transmissão de bits normais por um canal de comunicação.

- **Camada de Enlace:** principal função é transformar um canal de transmissão normal em uma linha que pareça livre de erros de transmissão.
- **Camada de Rede:** Controla a operação da sub-rede, determina como os pacotes são roteados da origem até o destino.
- **Camada de Transporte:** aceitar dados da camada acima dela, dividi-los em unidades menores, se for preciso, repassar essas unidades a camada de rede e garantir que todos os fragmentos chegarão corretamente à outra extremidade. Determina também o tipo de serviço.
- **Camada de Sessão:** Controle de diálogo, gerenciamento de *tokens*, sincronização.
- **Camada de Apresentação:** Sintaxe e semântica das informações, comunicação de computadores com diferentes representações.
- **Camada de Aplicação:** Contém uma série de protocolos comumente necessários para os usuários. Ex. HTTP.

Quais eram os problemas do padrão ISO/OSI?? Por que ele não deu certo??

Momento Ruim: O apocalipse dos dois elefantes: Fig. 1.21



Tecnologia Ruim: Escolha de 7 camadas foi mais política que técnica. Difícil implementação para equipamentos.
Implementações Ruins: As implementações realizadas se mostraram lentas. Enquanto que UNIX surgia com a pilha TCP/IP já implementada para ARPANET.
Política Ruim: em 80 as universidades tinha adoração por UNIX o que dificultou o uso do padrão.

O Modelo de referência TCP/IP: modelo com 4 camadas usada na avó de todas as redes a ARPANET.

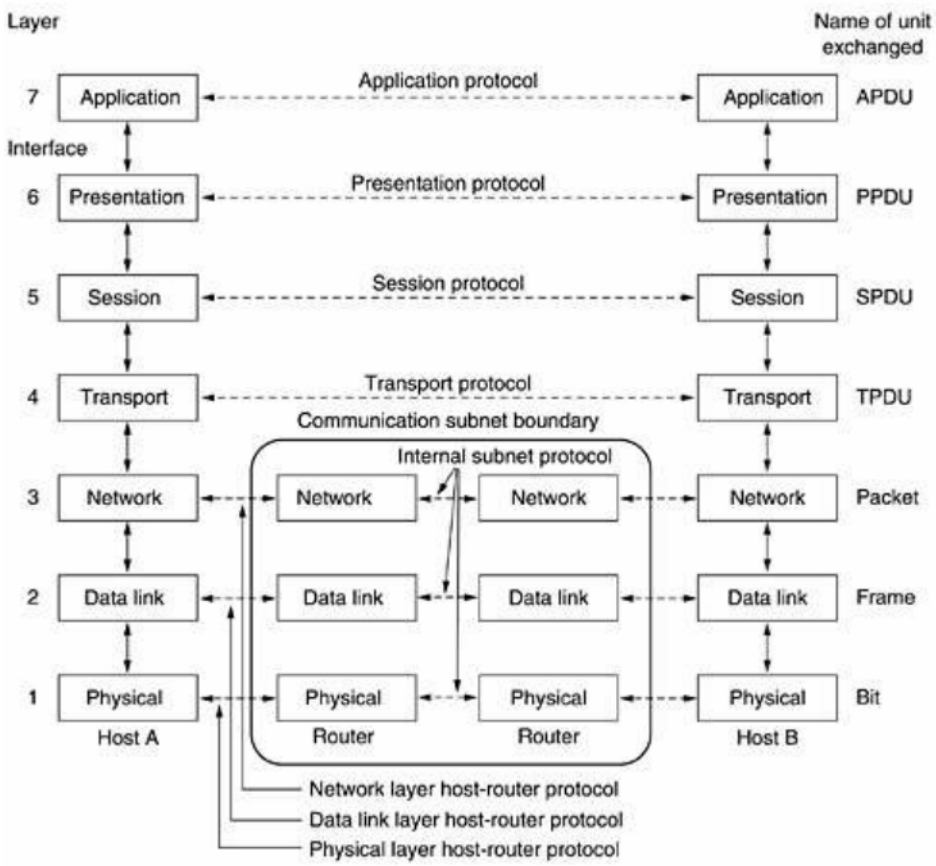
- **Camada de Enlace:** Camada de interconexão com serviço não orientado a conexão.
- **Camada de INTERNET (Camada de Rede):** integra toda a arquitetura, mantendo-a unida, correspondente a camada de rede do modelo OSI.
- **Camada de Transporte:** permitir que as entidades pares dos hosts de orgiem e de destino mantenham uma conversação igual modelo OSI. Porém com uso de protocolos específicos: TCP e UDP.
- **Camada de Aplicação:** Contém todos os protocolos de nível mais alto. HTTP, FTP, TELNET, RTP, SMTP.

Iremos aprender conforme segue no livro texto da disciplina o modelo de camadas: Física, Enlace, Rede, Transporte, Aplicação.

Parte 3: A Camada Física

1. A Camada Física:

É a camada mais baixa na hierarquia do modelo de protocolo. Ela define a interface elétrica, de sincronização e outras, pelas quais os *bits* são enviados como sinais pelos canais. A camada física é o alicerce sobre o qual a rede é construída. Como as propriedades dos diferentes tipos de canais físicos determinam o desempenho (por exemplo, *troughput*, latência e taxa de erros).
A camada 1 do **Modelo ISO/OSI** conforme podemos ver abaixo "*Physical*"**.**



2. Cabeamento Estruturado:

Pares Trançados: Um dos meios de transmissão mais antigos e ainda mais comuns é o par trançado. Um par trançado consiste de em dois fios de cobres encapados, que em geral tem cerca de 1 mm de espessura. Os fio enrolados de forma helicoidal, assim como DNA. O par trançado é feito porque dois fios paralelos formam uma antena simples, juntos apresentam menor interferência.

O cabeamento de rede mais comum é o **Categoria 5 (CAT 5),** apresenta-se com quatro pares de dois fios isolados e levemente trançados agrupados em uma capa plástica denominado conector RJ 45.
Diferentes padrões de LAN podem ser usados com os pares trançados de maneira diferente. Por exemplo, 100 MBps usa dois dos quatro pares, já *Gigabit* utiliza os quatro pares.

TIPO	USO	
Categoria 1	Voz (Cabo Telefônico)	São utilizados por equipamentos de telecomunicação e não devem ser usados para uma rede local
Categoria 2	Dados a 4 Mbps (LocalTalk)	
Categoria 3	Transmissão de até 16 MHz. Dados a 10 Mbps (Ethernet)	
Categoria 4	Transmissão de até 20 MHz. Dados a 20 Mbps (16 Mbps Token Ring)	
Categoria 5	Transmissão de até 100 MHz. Dados a 100 Mbps (Fast Ethernet)	

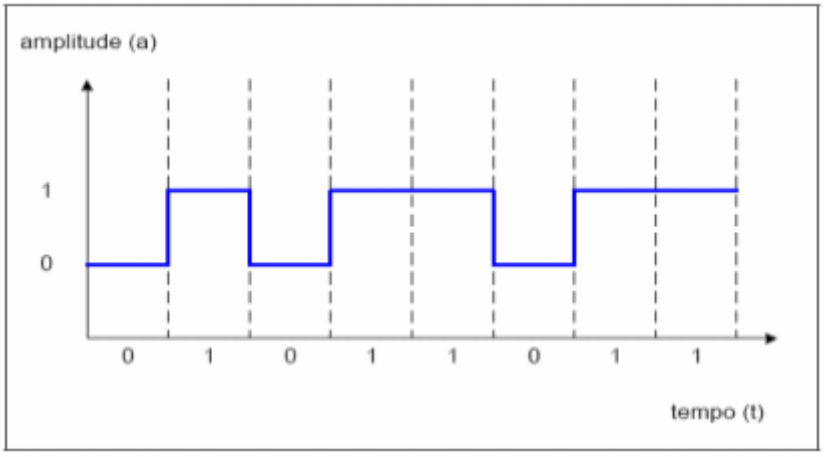
Sentido de envio de dados: FULL DUPLEX: enlace usados nos dois sentidos ao mesmo tempo como uma estrada de mão dupla. HALF DUPLEX: usados nos dois sentidos porém um de cada vez, como uma linha férrea com apenas um trilho.SIMPLEX: Tráfego apenas em uma direção.
Até a categoria 6, esses tipos de fios são conhecidos como par trançado não blindado ou **UTP (*Unshielded Twisted Pair*)**.

2. Transmissão de Sinal

A transmissão de sinal em uma rede de computadores é a propagação de ondas através de um meio físico (ar, fios metálicos, fibra de vidro) que podem ter suas características (amplitude, freqüência, fase) alteradas no tempo para refletir a codificação da informação transmitida. Esta informação está associada, em geral, às idéias ou dados manipulados pelos agentes que as criam, manipulam e processam. Neste cenário, os sinais correspondem à materialização específica dessas informações utilizada no momento da transmissão. Ainda, neste contexto e importante entender conceito de sinal analógico e sinal digital. O sinal analógico é um tipo de sinal contínuo que varia em função do tempo.

Figura 1 - Sinal Analógico

O sinal digital é uma seqüência de pulsos com amplitude fixa (em valores discretos), onde o sinal é construído através de uma seqüência de intervalos de tamanho igual a T segundos, chamados intervalos de sinalização. Um aspecto que pode comprometer a qualidade do sinal transmitido são os ruídos. Em qualquer transmissão, o sinal recebido é sempre igual ao sinal transmitido modificado



por distorções impostas por meios físicos e por distorções inseridas através de interferências indesejáveis ou ruídos (maior limitação no desempenho dos sistemas de comunicação). O ruído é medido pela razão entre a potência do sinal e a potência do ruído, chamada de razão (ou relação) sinal-ruído, medido por decibéis.
O Ruído pode ser:

- **Térmico:** causado pela agitação dos elétrons nos condutores, presente em todos os dispositivos eletrônicos e meios de transmissão, sendo uniformemente distribuído em todas as freqüências do espectro (ruído branco) com quantidade definida em função da temperatura.
- **Intermodular:** causado pelo compartilhamento de um mesmo meio físico (através de multiplexação de freqüência) por sinais de diferentes freqüências. Ocorre em geral devido a defeitos de equipamento ou na presença de sinais de potência muito alta.
- ***Crosstalk:*** causado pela interferência indesejável entre condutores muito próximos que induzem sinais entre si (linhas telefônicas cruzadas, cabos de pares trançados em redes Ethernet, por exemplo).
- **Impulsivo:** pulsos irregulares de grande amplitude, não contínuos e de difícil prevenção. Tem origem em várias fontes: distúrbios elétricos externos, falha de equipamento, etc. Na transmissão analógica, sendo de curta duração, não causam danos. Na transmissão digital são a maior causa de erros. A atenuação também é outro fator comprometedor nas transmissões. Caracteriza-se pela queda de potência de um sinal em função da distância de transmissão e do meio físico. De igual forma o eco, que é a reflexão de sinal quando há mudança da impedância (resistência à passagem de um sinal alternado) do meio de transmissão.

3. Tipos de meio físico para comunicação de dados:

3.1 Cabo par-trançado: Cabo Par Trançado (10 *BaseT*). Os cabos de pares trançados, um, dois ou quatro pares de fios são enrolados em espiral dois a dois de forma a reduzir o ruído e manter constantes as propriedades elétricas do meio ao longo de todo o seu comprimento. O trancamento inibe a interferência entre os pares (diafonia). Suporta transmissão analógica e digital, tem largura de banda relativamente alta (10/100/1000 Mbps, dependendo da distância, técnica de transmissão e qualidade do cabo).
Os cabos de pares trançados podem ser:
- Não blindado: (*Unshielded Twisted Pair - UTP*): quando seus pares são envolvidos unicamente por uma cobertura plástica (possuem preços mais econômicos, porém sujeitos às interferências). Normalmente 4 pares e bitola de 24 AWG (0,51 mm). Como não existe proteção estão sujeitos a interferências eletromagnéticas externas.

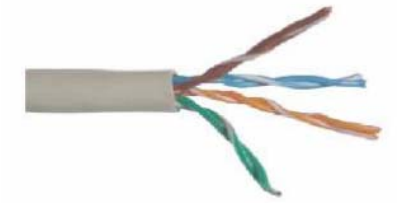


Figura 8 - Cabo UTP

- Blindado: (*Shielded Twisted Pair - STP*): quando seus pares são envolvidos por uma capa metálica (blindagem) e uma cobertura plástica. A malha metálica confere uma imunidade bastante boa em relação ao ruído, particularmente ao efeito *crosstalk* de fiações adjacentes.

a) Categoria 3:

- Pares trançados sólidos AWG 24;
- Impedância de 100 *Ohms*;
- Testado a 16 MHz para atenuação e paradiafonia;
- Padrão mínimo para 10BaseT.

b) Categoria 5:
- Pares trançados AWG 22 (rígido) ou AWG 24 (flexível);
- Impedância de 100 *Ohms*;
- Testado para a largura de 100 MHz;
- Pode ser usado para taxas de 100 Mpbs.

c) Categoria 5e:
- É uma melhoria da Categoria 5 (*enhanced*);
- Pode ser usado em redes *Gigabit*, 1000BaseT, com 4 pares.

d) Categoria 6:
- É compatível com a Categoria 5e;
- Melhor desempenho;
- Largura de banda de 250 MHz;
- Permite suporte a novas tecnologias como a *Ethernet* 10 Gbps sem investimentos adicionais na infra-estrutura atual.

	Largura de Banda mhz	Data Speed bps	
CAT 1	1	20 K	Analog Voice
CAT 2	4	1 M	Digital Voice
CAT 3	16	10 M	Ethernet
CAT 4	20	16 M	Ethernet
CAT 5	100	100 M	Fast Ethernet
CAT 5e	100	1 G (100m)	Gigabit Ethernet
CAT 6	250	2,5 G (100m)	Gigabit Ethernet
CAT 7	600	10 G (100m)	Gigabit Ethernet

A Conectorização do par trançado:

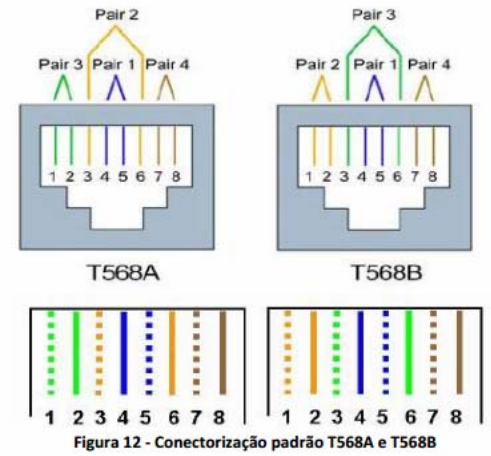


Figura 12 - Conectorização padrão T568A e T568B

3.2 Comunicação por linhas de energia elétrica.

As redes de telefonia e televisão não são as únicas fontes de fiação que podem ser reaproveitadas para comunicação. Há outro tipo de fiação que pode ser utilizada para tal propósito a rede de fiação elétrica. Por exemplo, o padrão X10 é capaz de enviar

dados em redes com pulsação de 60 MHz (Usina Itaipú). Mais utilizado tal padrão para sistema de domótica (automação residencial).

3.3 Fibra Ótica

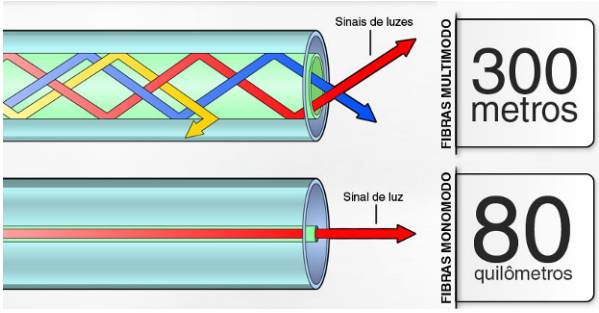
Custo alto de instalação porém alta velocidade de envio ordem de 10 Gbps. A fibra ótica é utilizada em transmissão por longa distância nos *backbones* da rede, LANs de alta velocidade. A grande vantagem de serem imunes a ruídos induzidos electromagneticamente.Dois tipos são encontrados para conexão: fibra de **multímodo e monomodo**,

- **Monomodo**

Como o nome já diz, as fibras monomodo só podem atender a um sinal por vez. Ou seja, uma única fonte de luz (na maior parte das vezes, laser) envia as informações por enormes distâncias. As fibras monomodo apresentam menos dispersão, por isso pode haver distâncias muito grandes entre retransmissores. Teoricamente, até 80 quilômetros podem separar dois transmissores, mas na prática eles são um pouco mais próximos. Outra vantagem das fibras desse tipo é a largura da banda oferecida, que garante velocidades maiores na troca de informações.

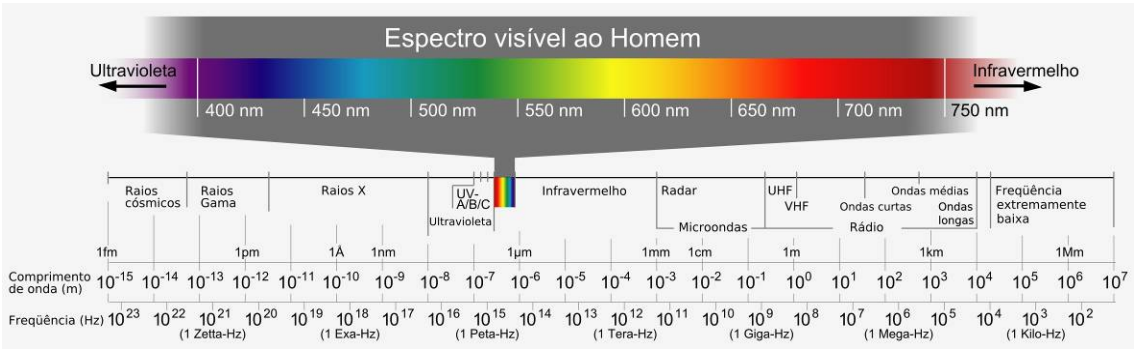
- **Multímodo**

Fibras multimodo garantem a emissão de vários sinais ao mesmo tempo (geralmente utilizam LEDs para a emissão). Esse tipo de fibra é mais recomendado para transmissões de curtas distâncias, pois garante apenas 300 metros de transmissões sem perdas. Elas são mais recomendadas para redes domésticas porque são muito mais baratas.



4. Comunicação Sem Fios

4.1 Espectro Eletromagnético: O espectro eletromagnético visual e não visual.



4.1.1 A Política do espectro Eletromagnético: Para evitar o caos têm sido feitos acordos nacionais e internacionais a respeito do uso de freqüências. As nações determinam alocação de faixas do espectro para AM, FM, TV, Celulares, telefonia,

polícia, militar, órgãos marítimos e governo. Em termos mundiais a ITU tenta coordenar essa alocação de forma para que sejam fabricados dispositivos que funcionem no mundo todo. De acordo com a proposta, a maioria dos governos reserva algumas bandas de freq., chamadas de ISM (INDUSTRIAL, SCIENTIFIC, MEDICAL) para uso sem licença. Sistema de garagens, telefones sem fio, microondas, etc. Tais equipamentos trabalham sobre potência máxima de 1 *Watt*.

As Faixas ISM são: 900 MHz, 2.4 GHz, 5.4 GHz e 5.8 GHz.

4.1.2 Reguladora no Brasil: ANATEL.

Mas como sair de interferências se diversos aparelhos trabalham nas mesmas freqüências?
"Tanto nos canais com fios e sem fios estamos sujeitos a interferências. Os canais sem fio e com fio transportam sinais analógicos, como a tensão variando continuamente, a intensidade da luz ou intensidade da onda. Para enviar informações temos que criar sinais digitais para representar os bits. O processo de conversão entre bits e sintais que os representam é chamado de **modulação digital.** Muitas vezes como necessitamos compartilhar o canal, ou espectro no caso das redes sem fio, é necessário realizar o que chamamos de **multiplexação.** A multiplexação pode ser realizada de diversas manteiras: divisão de tempo, freqüência e código."

4.2 Espectro por salto de freqüência: o transmissor salta de uma freq. para outra centenas de vezes por segundo. Essa robustez torna a técnica útil para as partes mais sobrecarregadas do espectro (2.4 GHz). Utilizada por Bluetooth e IEEE 802.11 b.

4.3 Multiplexação por divisão de freq ou FDM (*Frequency Division Multiplexing*): Separação de frequências para envio com posse exclusiva de cada canal. Utilizado em AM/FM. Na multiplexação ortogonal por divisão de fequência, ou OFDM (*Orthogonal Frequency Divsion Multiplexing*), a largura de banda do canal é dividida em muitas subportadoras que enviam dados independentemente. Utilizado em IEEE 802.11 e pesquisado para 4G. Em linhas gerais enviam as informações em freqüências sobrepostas porém sem interferência. Então a FFT pode remontar tais freqs. OFDM usado em IEEE 802.16d.

4.4 Multiplexação por Dvisão de Tempo: TDM (*Time Divsion Multiplexing*): Os usuários alternam-se num padrão de rodízio cada um periodicamente usando a largura de banda inteira por um período curto de tempo.Os bits de cada fluxo de entrada são separados em **slots de tempo.**

4.5 Multiplexação por Divisão de Código: Espectro de dispersão de sequência direta (CDMA Code Division Multiple Access): Usa uma sequência de código para disperar o sinal de dados por uma banda de freq. mais ampla. O CDMA permite que cada estação transmita por todo o espectro de freq. o tempo todo. A chave do CDMA é extrair sinal desejado e rejeitar todos os outros de ruído aleatório.

Explicação Didática: O saguão do aeroporto com muitos pares de pessoas conversando ao mesmo tempo. TDM: todas as pessoas estariam no meio do salão porém conversariam por turnos. Com a FDM, as pessoas formariam grupos bem separados, cada uma mantendo sua conversação ao mesmo tempo, alguns com uma altura maior e outros com uma altura menor. Já CDMA ,todas as pessoas estariam no meio do saguão

falando ao mesmo tempo mas cada par de pessoas conversando em um idioma diferente. O par que falasse indiano somente entenderia outro par que por ventura estivesse falando indiano também. Outra explicação didática refere-se ao chuveiro elétrico e ao fluxo contínuo de uma mangueira. FDM (fluxo contínuo de água) e OFDM (chuveiro elétrico)

4.6 Telefonia Celular

A primeira geração dos celulares era analógica **(voz analógica)**, a segunda geração digital. A troca para digital tem diversas vantagens. Ela oferece ganhos de capacidade, permitindo que os sinais de voz sejam digitalizados e compactados. Melhor segurança, etc.

4.6.1 GSM (*Global System for móbile communications*) – voz digital, surgiu em 1991 como esforço para produzir o padrão 2G europeu. Mantém do 1G a caracterização de células, reutilização de freqs. das células não vizinhas e mobilidade com *handoffs* à medida que os assinantes se movem. Utilização de *chip* removível (*SIM – Subscriber Identity Module* – Módulo de identidade do assinante). GSM trabalha em faixa de freqs. de 900, 1800 e 1900 MHz. Sistema celular **duplex** por divisão de freqüência.Contudo um único par de freqs. é dividido por multiplexação por divisão em *slots* de tempo. Desse modo, é compartilhado a freq. por vários aparelhos.

4.6.2 3G: Voz e dados Digitais: Problemas encontrados atualmente no setor: Tráfego de dados está superando o tráfego de voz na rede. Atualmente dispositivo não é mais apenas para voz, web, redes sociais, email, multimídia, etc.. Com o surgimento do iOS e Android isso se intensificou. Como conseguir o envio de dados e voz ao mesmo tempo basicamente com o mesmo espectro e dividi-lo? Resposta: CDMA aperfeiçoado como utilizado EDGE.

4.6.3 4G: Está definido?? Brasil = Negativo: WiMAX ou LTE briga política ainda. Porém os sistemas de TV Digital que necessitam do canal de retorno provavelmente sairão do papel quando ocorrer tal desenvolvimento.

Parte 4: A Camada de Enlace

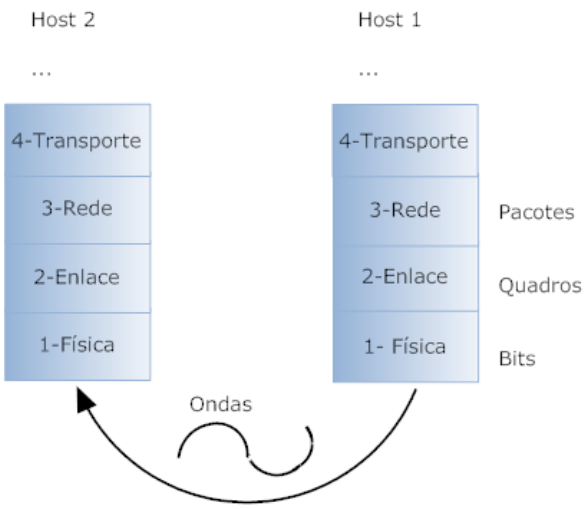
1. A Camada de Enlace de Dados

Infelizmente muitas vezes os canais de comunicação produzem erros e necessitamos entender como corrigi-los para criar viável uma comunicação eficaz. Existem limitações relacionadas ao tamanho do canal, atrasos no envio e recebimento das informações, alteração constante da taxa de dados no envio, e ordem diferente de recebinento.

Funções da Camada de Enlace de Dados:

1. Fornecer uma interface de serviço bem definida à camada de rede;
2. Lidar com Erros de Transmissão;
3. Regular o fluxo de dados de tal forma que receptores lentos não sejam atropelados por transmissores rápidos.

Para corrigir esses erros a camada de enlace encapsula esses dados dos pacotes da camada de rede em **quadros** para transmissão pelo meio físico**.** Cada quadro contém um cabeçalho (*header*), um campo de carga útil (*payload*) e um final (*trailer*).



A camada de enlace pode ser projetada de modo a oferecer diversos serviços.

2 Tipos de Serviços:

Os serviços reais oferecidos podem variar de um protocolo para outro. 3 possibilidades de funcionamento para tais protocolos:

- **Serviço não orientado a conexões sem confirmação:** consiste em fazer a máquina de origem enviar quadros independentes à máquina de destino, sem que esta confirme o recebimento desses quadros. A Ethernet é um bom exemplo deste tipo de serviço. Se algum for perdido em decorrência de ruídos na linha, não haverá nenhuma tentativa de detectar a perda ou recuperá-la na camada de enlace! Ficando a cargo de outra camada.
- **Serviço não orientado a conexões com confirmação:** Cada quadro enviado é confirmado individualmente. O transmissor sabe se um quadro chegou corretamente ou não. Caso não tenha chegado dentro de um intervalo específico, o quadro poderá ser enviado outra vez. Útil em canais não confiáveis ex. IEEE 802.11.
- **Serviço orientado a conexões com confirmação:** As máquinas de origem e destino estabelecem uma conexão antes de qualquer dado ser transferido. Cada quadro enviado pela conexão é numerado, e a camada de enlace garante que cada quadro será, de fato, recebido. Útil para enlaces longos como satélites e circuito telefônico interubano. Se o serviço não orientado a conexões com confirmação fosse usado é possível imaginar que as confirmações perdidas poderiam fazer com que um quadro inteiro fosse enviado e recebido várias vezes, desperdiçando banda.

2.1 Enquadramento

Para oferecer serviços a camada de rede a camada de enlace deve usar o serviço fornecido pela camada física. O que a camada física faz é aceitar um fluxo de *bits* brutos e tentar entregá-lo ao destino. Se o canal tiver ruído, como acontece na maioria dos canais sem fio e alguns com fio, a camada física acrescentará alguma redundância aos seus sinais, para reduzir a taxa de erros de *bits* para um nível tolerável. Contudo, o fluxo de *bits* recebido pela camada de enlace não tem garantia de estar livre de erros. Alguns *bits* podem ter valores diferentes e o número de *bits* recebidos pode ser menor, igual ou maior que o número de *bits* transmitidos. A camada de enlace de dados é responsável por detectar e, se necessário, corrigir tais erros. Em geral a estratégia utilizada pela

camada é dividir o fluxo de bits em quadros distintos, calcular um pequeno valor (*token*), chamado de *checksum* (somatório de verificação), para cada quadro e incluir essa soma de verificação no quadro quando ele for transmitido. Quando um quadro chega a seu destino o *checksum* é recalculado. Se o *checksum* recém-calculado for diferente for diferente do contido no quadro, a camada de enlace saberá que houve um erro. Quatro métodos são possíveis para realizar tal ação:

- **Contagem de caracteres:** utiliza um campo no cabeçalho para especificar o n° de *bytes* no quadro. Quando vê a contagem de caracteres, a camada de enlace de dados de destino sabe quantos bytes devem vir em seguida e, conseqüentemente, onde está o fim do quadro. O problema com esse algoritmo é que a contagem pode ser adulterada por um erro de transmissão. Não é muito usado Figura 3.3(a,b).
- **Bytes de flag com inserção de bytes:** Contorna o erro anterior fazendo com cada quadro comece com bytes especiais. Normalmente o mesmo *byte*, chamado de *byte* de *flag*.
- **Flags iniciais e finais, com inserção de bits:** Com cada quadro comece e termine com bytes especiais. Normalmente o mesmo byte, chamado de byte de flag. Dois bytes de flag indicam o fim de um quadro e início de outro. Assim, se o receptor perder a sincronização, ele poderá simplesmente procurar dois bytes de flag para encontrar o final do quadro atual e o início do seguinte. Problema: Se aparecer como em imagens digitais o caracter especial no meio do quadro? Por exemplo no *payload*.
- **Violações de codificação da camada física:** Atalho da camada física, foco em encontrar o início e o final dos quadros. Um padrão muito comum em IEEE 802.11 é fazer com que um quadro comece com um padrão bem definido chamado *preâmbulo*. Esse padrão pode ser muito longo no caso de IEEE 802.11 (72 bits). O preâmbulo é, então, seguido por um campo de comprimento (contador) no cabeçalho, que é usado para localizar o final do quadro.

Flag 01111110	Quadro	Flag 01111110

Flag 01111110	Endereço	Controle	Protocolo	Dados	CDE	Flag 01111110

Figura 5 – Delimitadores

O grande problema desta solução é quando a informação transmitida incorrer na mesma representação do *flag*. A solução para este problema são as técnicas de ***byte stuffing*** e ***bit stuffing.*** Ambas as técnicas consistem na utilização de um *scape*. Veja:

Quadro Original		
FI	FF ... FI ... FF ... CECE	FF

Quadro Transmitido		
FI	**CE**FF ... **CE**FI ... **CE**FF ... **CE**C**E**C**E**CE	FF

Figura 6 - *Byte Stuffing*

No *Bit Stuffing* a funcionalidade é parecida, porém para quebrar a igualdade a cada 5 bits 1 é inserido um 0. Ao ser recebido pelo receptor esse 0 é retirado e se aparecer um 0 após uma seqüência de 5 bits 1, significa que a seqüência é uma informação.

Quadro Original		
01111110	0111111010110	01111110

Quadro Transmitido		
01111110	011111**0**1010110	01111110

Quadro Recebido		
01111110	011111**0**1010110	01111110

Quadro Original		
01111110	0111111010110	01111110

Figura 7 - *Bit Stuffing*

Além dessas soluções, existem outras alternativas que levam em consideração o tamanho do quadro.

2.2 Controle de Fluxo da Camada de Enlace

Como resolver o problema de envio de dados de um receptor lento e um transmissor rápido? Como no caso dos smartphones, mesmo que a conexão não tenha perdas os aparelhos não tem grande capacidade de recebimento de dados, independentemente do canal de comunicação, devido ao seu processamento e S.O. Duas maneiras são usadas para realizar tal procedimento de validação dos dados para esse tipo de enlace. A primeira refere-se ao *controle de fluxo baseado em feedback* onde o receptor envia ao transmissor sua capacidade de recebimento para que o emissor não transborde o canal e percam-se os dados. O segundo é baseado no *controle de fluxo baseado na velocidade.* Tem-se um protocolo limitador de velocidade de dados do emissor para o receptor.

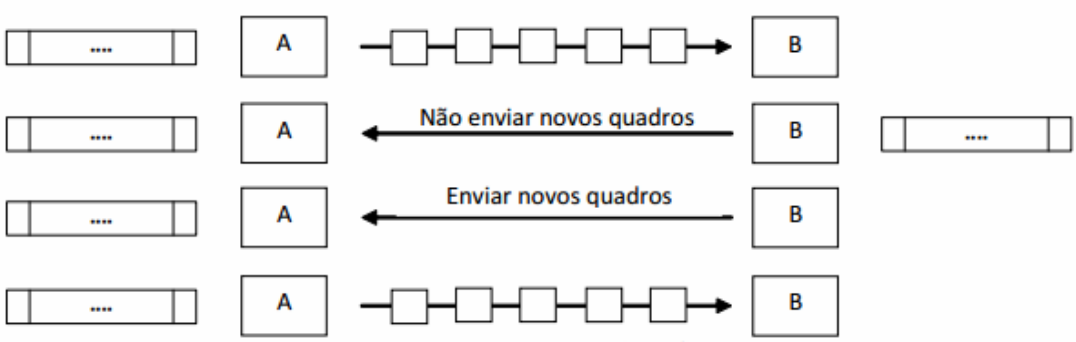


Figura 11 – Controle de fluxo

2.3 Controle de Erros da Camada de Enlace

Após resolvermos os problemas de delimitação de início e fim de cada quadro, vamos ao seguinte problema: como ter certeza de que todos os quadros serão entregues a camada de rede de destino e na ordem apropriada? Suponha, para o momento, que o receptor possa saber se um quadro que ele recebe contém informações corretas ou defeituosas. Para serviços não orientados a conexão, sem confirmação, pode ser suficiente que o emissor apenas continue enviando quadros sem se importar se eles chegaram corretamente, mas sem dúvida essa não seria uma boa opção para serviços orientados a conexões confiáveis. A forma mais comum de garantir uma entrega confiável é dar ao transmissor algum tipo de feedback sobre o que está acontecendo no outro extremo da linha. Normalmente, o protocolo solicita que o receptor retorne quadros de controle especiais com confirmações positivas ou negativas sobre os quadros recebidos. Se for positiva, saberá que o quadro chegou com segurança, caso contrário houve algum problema na conexão e deve ser retransmitido.

Complicação adicional 1: se houver uma rajada de ruídos? E o receptor nem saber que lhe estão enviando algo? Nesse caso o receptor não reagirá solicitando o reenvio pois nem saberá que algo foi enviado.

Complicação adicional 2: E se o quadro de confirmação se perder? O emisor não saberá como prosseguir. Envia novamente ?

Como tratar isso? 1-) Uma possibilidade é adicionando *timers* suficientemente grandes para que dê tempo o envio, recebimento pelo receptor, tratamento e resposta chegando até o emissor. Caso o *timer* estoure é enviado novamente o quadro. 2-) Outra possibilidade para tratar o problema do envio duplo de quadros e repassamento duplo a camada de rede pelo receptor é numerar os quadros no envio para que o receptor saiba que os mesmos são diferentes.

2.4 Código de Correções de Erros: Todos esses códigos adicionam redundância às informações enviadas.

Hamming: Para entender como os erros podem ser tratados, é necessário verificar de perto o que é de fato um erro. Normalmente, um quadro consiste em m bits de dados (ou seja, de mensagens) e de r bits redundantes ou de verificação. Seja o tamanho total n (isto é, n = m + r). Com freqüência, uma unidade de n bits que contém bits de dados e bits de verificação é chamada palavra de código (*codeword*) de n bits. Dadas duas palavras de código, digamos 10001001 e 10110001, é possível determinar quantos bits correspondentes apresentam diferenças. Nesse caso, são 3 os *bits* divergentes. Para determinar quantos *bits* apresentam diferenças, basta efetuar uma operação OR exclusivo entre as duas palavras de código, e contar o número de *bits* 1 no resultado. Por exemplo:

```
10001001
10110001
00111000
```

O número de posições de bits em que duas palavras de código diferem entre si é chamado distância de *Hamming* (*Hamming*, 1950). Isso significa que, se duas palavras de código estiverem a uma distância de *Hamming* igual a d uma a outra, será necessário corrigir d erros de bits isolados para converter uma palavra na outra. Como um exemplo simples de código de correção de erros, considere um código contendo apenas quatro palavras de código válidas:

0000000000, 0000011111, 1111100000 e 1111111111

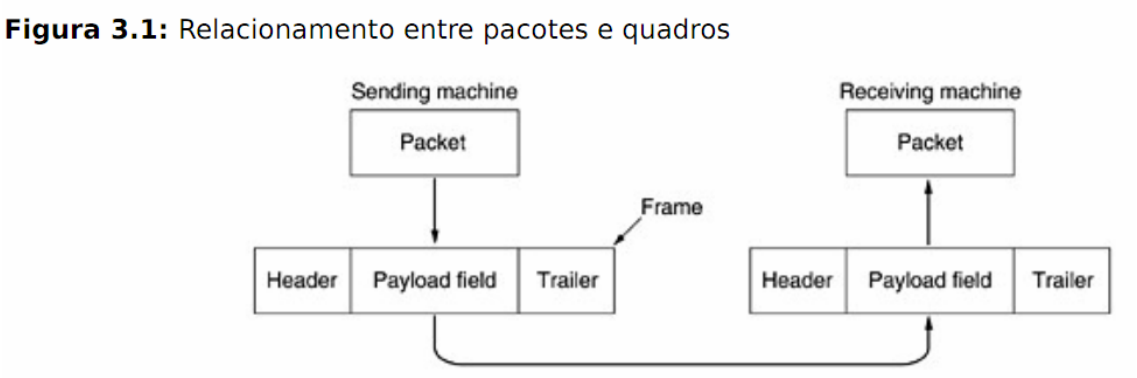
Esse código tem uma distância igual a 5, o que significa que ele pode corrigir erros duplos. Se a palavra de código 0000000111 for detectada, o receptor saberá que a original deve ter sido 0000011111. No entanto, se um erro triplo transformar 0000000000 em 0000000111, o erro não será corrigido da maneira adequada. Suponha que desejamos criar um código com m bits de mensagem e r bits de verificação que permitirão a correção de todos os erros simples. Cada uma das 2m mensagens válidas tem n palavras de código inválidas a uma distância igual a 1 da mensagem. Essas palavras inválidas são formadas pela inversão sistemática de cada um dos n bits da palavra de código de n bits formada a partir dela. Portanto, cada uma das 2m mensagens válidas exige n + 1 padrões de bits dedicados a ela. Como o número total de padrões de bits é 2n , devemos ter $(n + 1)2m \leq 2n$. . Utilizando n = m + r, esse requisito passa a ser $(m + r + 1) \leq 2r$. Se m for determinado, o limite para o número de bits de verificação necessários para corrigir erros isolados será mais baixo.

3. Protocolos da Camada de Enlace

No que se refere à camada de enlace de dados, o pacote repassado a ela pela camada de rede através da interface consiste em dados puros, em que cada bit deve ser entregue à camada de rede de destino. O fato de a camada de rede de destino poder interpretar parte do pacote como um cabeçalho não tem nenhum interesse para a camada de enlace de dados.

Quando a camada de enlace de dados aceita um pacote, ela o encapsula em um quadro, acrescentando-lhe um cabeçalho e um final de enlace de dados (veja a Figura 3.1).

Figura 3.1: Relacionamento entre pacotes e quadros



Portanto, um quadro consiste em um pacote incorporado, em algumas informações de controle (no cabeçalho) e em um total de verificação (no final). Em seguida, o quadro é transmitido à camada de enlace de dados da outra máquina. Presumiremos que existem procedimentos de biblioteca adequados, para enviar um quadro e *from_physical_layer* para receber um quadro. O *hardware* de transmissão calcula e acrescenta o total de verificação (criando assim o final), de forma que o software da camada de enlace de dados não precise se preocupar com isso.

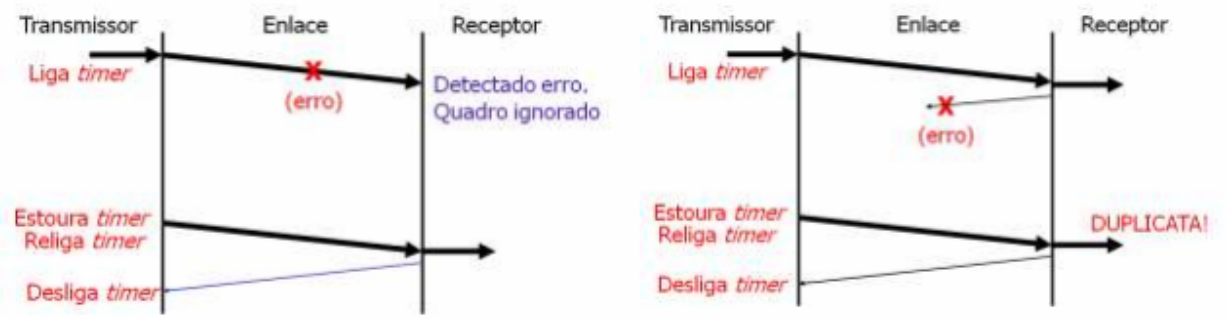
3.1 Um protocolo *simplex* sem restrição: É um protocolo muito simples, onde os dados são transmitidos somente em um sentido. As camadas de rede do transmissor e do receptor estão sempre prontas à espera de informações. O tempo de processamento pode ser ignorado, o espaço em *buffer* é infinito e o canal de comunicação entre as camadas de enlace de dados nunca é danificado e nem perde quadros. Em suma, é um protocolo imaginário ou “utópico”. O protocolo consiste em dois procedimentos distintos, um que envia e outro que recebe a informação. Neste caso, não são usados números de seqüência ou de confirmação. A parte referente aos dados é repassada à camada de rede, e a camada de enlace de dados volta a esperar pelo próximo quadro, ficando efetivamente em suspenso até a chegada de outro quadro.

3.2 Um protocolo *simplex stop-and-wait*: O principal problema que este protocolo veio lidar foi impedir que o transmissor inunde o receptor com dados, mais rapidamente do que este é capaz de processá-los. Em determinadas circunstâncias talvez seja possível para o transmissor simplesmente inserir um retardo no protocolo 1, a fim de reduzir sua velocidade e impedi-lo de sobrecarregar o receptor. Uma solução mais viável é fazer o receptor enviar um *feedback* ao transmissor, ou seja, uma vez enviado um determinado quadro, outro somente será enviado após o recebimento de uma confirmação. Esta estratégia é um mecanismo, inclusive de controle de fluxo. Embora o tráfego de dados seja simplex, há fluxo de quadros em ambos os sentidos.

3.3 Um protocolo *simplex* com canal de ruído: Neste protocolo considera-se uma situação normal em um canal de comunicação, no qual ocorrem erros. Os quadros podem ser danificados ou completamente perdidos. Se assim acontecer, supomos que o hardware receptor detectará essa ocorrência ao calcular o total de verificação.Somente uma confirmação por parte do receptor não é suficiente. Caso a comunicação seja perdida, em muitos casos é necessário realizar uma retransmissão, e para isso é indispensável adicionar um número de seqüência no cabeçalho de cada quadro enviado de modo que o receptor saiba que o quadro enviado novamente é o mesmo que fora enviado anteriormente. O receptor informa caso recepção corra sem problemas e é

importante notar que o número de seqüência pode ter comprimento de apenas 1 bit. Esse tipo de protocolo tem diversas variantes, a saber:

- **PAR** (*Positive Acknowlodgement with Retransmission*) – Confirmação Positiva com Retransmissão;
- **ARQ** (*Automatic Repeat reQuest*) – Solicitação de Repetição Automática.



3.4 Protocolos de Janela deslizante O protocolo de janelas deslizantes é usado para a entrega confiável e ordenada de mensagens. É um protocolo orientado a conexão (primeiro garante que a conexão está ativa, para depois iniciar o envio das mensagens) que garante que todas as mensagens enviadas são entregues aos destinatários integralmente e na ordem correta de envio. O receptor envia uma mensagem de confirmação de recebimento (ACK) a cada mensagem recebida. Se o transmissor não recebe o ACK de uma mensagem num tempo pré-estabelecido, ele envia a mesma mensagem novamente. O transmissor cria uma espécie de tabela, onde cada posição é uma janela, em que são gravadas todas as mensagens que foram enviadas.

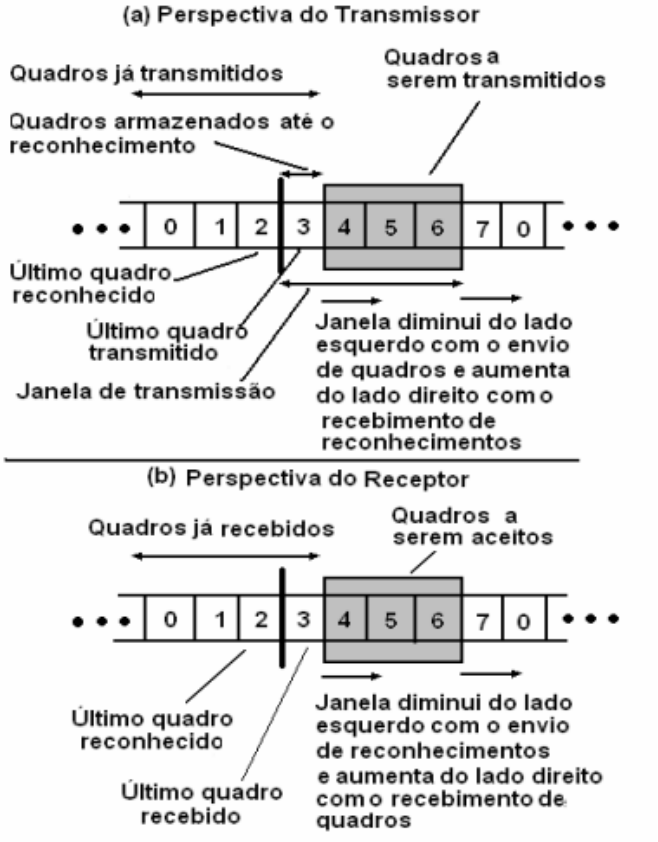


Figura 15 - Protocolo de Janelas Deslizantes

4. A Camada de Enlace da Internet:

A Internet consiste em máquinas individuais (*hosts* e roteadores) e na infra-estrutura de comunicação que as conecta. Na prática, a comunicação ponto-a-ponto é utilizada principalmente em duas situações. Na primeira, milhares de organizações têm uma LAN ou mais e um roteador ou ponte (*Bridge*). Comumente os roteadores são interconectados por uma LAN de *backbone*. Assim, todas as conexões como mundo exterior passam por um ou mais roteadores que têm linhas privativas ponto-a-ponto. São esses roteadores e suas linhas que compõem as sub-redes de comunicação que a *Internet* se baseia. Os computadores domésticos estabelecem uma conexão com um determinado roteador de um provedor de serviços da *Internet*.

Parte 5: A Sub-Camada de Acesso ao Meio

---

1. A Sub-Camada de Acesso ao Meio

Em qualquer rede de difusão, a questão fundamental é determinar quem tem direito de usar o canal quando há uma disputa por ele. Para tornar essa questão mais clara, considere uma chamada de teleconferência, na qual seis pessoas em seis diferentes telefones estão todas conectadas entre si, de forma que cada uma pode ouvir e falar com todas as outras. É muito provável que, quando uma delas parar de falar, duas ou mais comecem a falar ao mesmo tempo, levando ao caos. Em uma reunião face a face, a confusão é evitada por meios externos. Por exemplo, em uma reunião, as pessoas levantam as mãos para pedir permissão para falar. Quando apenas um único canal está disponível, a determinação de quem deve ser o próximo a falar é muito mais difícil. Existem vários protocolos destinados a solucionar o problema, e eles formam o conteúdo deste capítulo. Na literatura, os canais de difusão às vezes são referidos como canais de multiacesso ou canais de acesso aleatório.

Os protocolos usados para determinar quem será o próximo em um canal de multiacesso pertencem a uma subcamada da camada de enlace de dados, chamada subcamada MAC (*Medium Access Control*). A subcamada MAC é especialmente importante em LANs que, em sua maioria, utilizam um canal de multiacesso como base de sua comunicação.

1.1 Alocação Estática de Canais: A maneira tradicional de alocar um único canal, tal como um tronco telefônico, entre vários usuários concorrentes é usar a FDM (*Frequency Division Multiplexing*)[1]. Se existem N usuários, a largura de banda é dividida em N partes do mesmo tamanho e a cada usuário será atribuída uma parte.

No entanto, quando o número de transmissores é grande e continuamente variável, ou quando o tráfego ocorre em rajadas, a FDM apresenta alguns problemas. Se o espectro for dividido em N áreas, e menos de N usuários estiverem interessados em estabelecer comunicação no momento, uma grande parte do espectro será desperdiçada. Se mais de N usuários quiserem se comunicar, alguns deles terão o acesso negado por falta de largura de banda, mesmo que alguns dos usuários aos quais foi alocada uma banda de freqüência raramente transmitam ou recebam dados.

Os mesmos argumentos que se aplicam à FDM também se aplicam à TDM (*Time Division Multiplexing*). Para cada usuário, é alocado estaticamente o N-ésimo slot de tempo. Se o usuário não empregar o slot alocado, este será

[1] ≈ Chuveiro ou AM/FM cada um tem o seu canal separadamente.

simplesmente desperdiçado. O mesmo é válido se dividirmos as redes fisicamente. Usando mais uma vez nosso exemplo anterior, se substituíssemos a rede de 100 Mbps por 10 redes de 10 Mbps cada uma e fizéssemos a alocação estática de cada usuário a uma delas, o retardo médio saltaria de 200 s para 2 ms. Como nenhum dos métodos estáticos tradicionais de alocação de canais funciona bem com um tráfego em rajadas, agora vamos tratar dos métodos dinâmicos.

1.2 Alocação Dinâmica de Canais: Premissas para a alocação dinâmica de Canais.

1. **Modelo da estação**. O modelo consiste em N estações independentes (computadores, telefones, comunicadores pessoais etc.), cada qual com um programa ou usuário que gera quadros para transmissão. Algumas vezes, as estações são chamadas terminais. A probabilidade de um quadro ser gerado em um intervalo de duração Δt é $\lambda \Delta t$, onde λ é uma constante (a taxa de chegada de novos quadros). Uma vez gerado um quadro, a estação é bloqueada e nada faz até que o quadro tenha sido transmitido com êxito.
2. **Premissa de canal único.** Um único canal está disponível para todas as comunicações. Todas as estações podem transmitir e receber por ele. No que se refere ao *hardware*, todas as estações são equivalentes, embora um *software* de protocolo possa atribuir prioridades a elas.
3. **Premissa de colisão.** Se dois quadros são transmitidos simultaneamente, eles se sobrepõem no tempo, e o sinal resultante é adulterado. Esse evento é denominado colisão. Todas as estações podem detectar colisões. Um quadro que tenha sofrido colisão terá de ser retransmitido posteriormente. Não há outros erros além dos gerados por colisões.
4. **Tempo contínuo e Tempo Segmentado (*Slotted*).** A transmissão por quadro pode começar a qualquer instante. Não há um relógio-mestre dividindo o tempo em intervalos discretos. No tempo segmentado (*slotted*). O tempo é dividido em intervalos discretos (*slots*). As transmissões de quadros sempre começam no início de um slot. O slot pode conter 0, 1 ou mais quadros, correspondentes a um *slot* ocisoso, uma transmissão bem-sucedida ou a uma colisão, respectivamente.
5. **Detecção de portadora (*carrier sense*) ou sem detecção de portadora.** As estações conseguem detectar se o canal está sendo usado antes de tentarem utilizá-lo. Se for detectado que o canal está ocupado, nenhuma estação tentará usá-lo até que ele fique livre. Quando não há detecção de portadora as estações não conseguem detectar o canal antes de tentar utilizá-lo. Elas simplesmente vão em frente e transmitem. Somente mais tarde conseguem determinar se a transmissão foi ou não bem-sucedida.

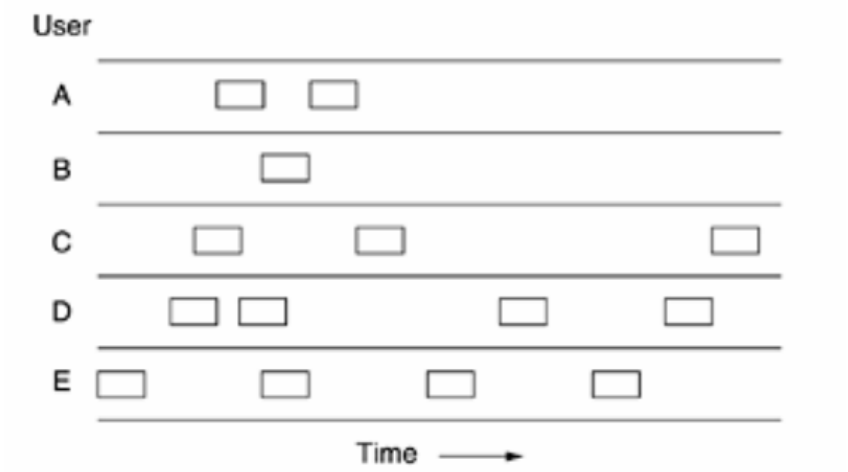
Ainda é necessário discutir essas premissas um pouco mais. A primeira diz que as estações são independentes, e que a carga é gerada a uma taxa constante. Há também a premissa de que cada estação tem apenas um programa ou usuário e, portanto, enquanto a estação estiver bloqueada, não será gerada qualquer nova carga. Os modelos mais sofisticados permitem estações multiprogramadas capazes de gerar mais carga enquanto uma estação está bloqueada, mas a análise dessas estações é muito mais complexa.

1.3 Protocolos de Acesso Múltiplo

1.3.1 ALOHA: Na década de 1970, Norman Abramson e seus colegas da Universidade do Havaí elaboraram um método novo e sofisticado para resolver o

problema de alocação de canais. Descreveremos aqui duas versões do ALOHA: puro e *slotted*. Elas diferem quanto ao fato de o tempo estar ou não dividido em *slots* discretos, nos quais todos os quadros devem se ajustar. Ao contrário do *slotted* ALOHA, o ALOHA puro não exige a sincronização de tempo global.

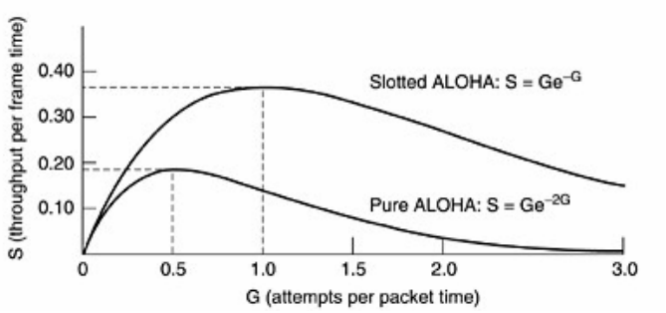
Puro: A idéia básica de um sistema ALOHA é simples: permitir que os usuários transmitam sempre que tiverem dados a ser enviados. Naturalmente, haverá colisões, e os quadros que colidirem serão danificados. Porém, devido à propriedade de feedback da difusão, um transmissor sempre consegue descobrir se seu quadro foi ou não destruído, da mesma maneira que o fazem outros usuários, bastando para isso escutar a saída do canal. Em uma LAN, esse feedback é imediato. Em um satélite, há uma demora de 270 ms antes de o transmissor saber se houve êxito na transmissão. Se não for possível por alguma razão realizar a escuta durante a transmissão, serão necessárias confirmações. Se o quadro foi destruído, o transmissor apenas espera um período de tempo aleatório e o envia novamente. O tempo de espera deve ser aleatório, pois senão os mesmos quadros continuarão a colidir repetidas vezes. Os sistemas em que vários usuários compartilham um canal comum de forma que possa gerar conflitos em geral são conhecidos como sistemas de disputa. A Figura 4.1, mostra um esboço da geração de quadros em um sistema ALOHA. Os quadros foram criados com o mesmo comprimento porque o *throughput*[2] dos sistemas ALOHA é maximizado quando o comprimento dos quadros é uniforme em vez de variável.



Sempre que dois quadros tentarem ocupar o canal ao mesmo tempo, haverá uma colisão e ambos serão danificados. Se o primeiro bit de um novo quadro se sobrepuser apenas ao último bit de um quadro quase terminado, os dois quadros serão totalmente destruídos e terão de ser retransmitidos posteriormente. O total de verificação não consegue (e não deve) fazer distinção entre uma perda total e uma perda parcial. Quadro com erro é quadro com erro, não há distinções.

1.3.2 Slotted: Em 1972, Roberts publicou um método para duplicar a capacidade de um sistema ALOHA (Roberts, 1972). Sua proposta era dividir o tempo em intervalos discretos, com cada intervalo correspondendo a um quadro. Esse método exige que os usuários concordem em relação às fronteiras dos slots. Uma forma de alcançar a sincronização entre os usuários seria ter uma estação especial que emitisse um sinal sonoro no início de cada intervalo, como um relógio. Figura 4.3: *Throughput* em comparação com o tráfego oferecido para sistemas ALOHA

[2] *Throughput*: Definição de taxa de transferência é a quantidade de dados transferidos de um lugar a outro.



No método de Roberts, que passou a ser conhecido como *slotted* ALOHA, em contraste com o ALOHA puro de Abramson, um computador não tem permissão para transmitir sempre que um caractere de retorno de cursor é digitado. Em vez disso, é necessário esperar o início do próximo *slot*. Quando o acesso à Internet por cabo foi criado, surgiu o problema de como alocar um canal compartilhado entre vários usuários concorrentes, e o *slotted* ALOHA foi resgatado para salvar a situação.

1.3.3 Protocolos de Acesso Múltiplo com Detecção de Portadora: Visando melhorar o desempenho do Aloha para acesso ao meio foram criados protocolos nos quais as estações escutam uma portadora (isto é, uma transmissão) e funcionam de acordo com ela são denominados protocolos com detecção de portadora (*carrier sense protocols*).

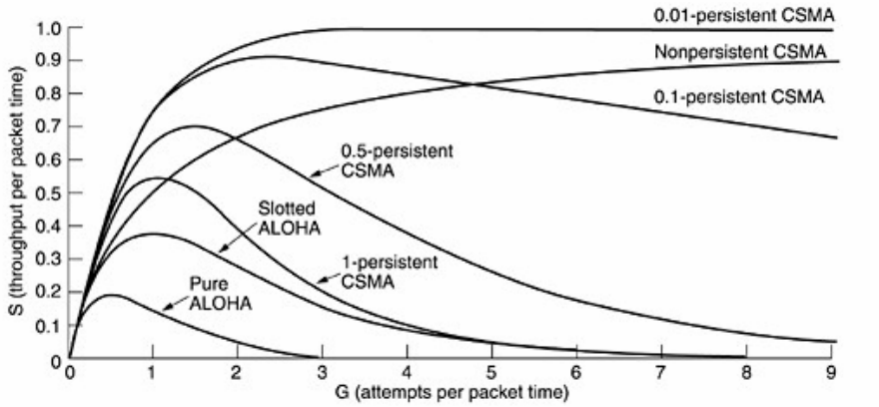
1.3.4 CSMA persistente e não persistente: O primeiro protocolo com detecção de portadora que estudaremos aqui denomina-se CSMA (*Carrier Sense Multiple Access*) 1-persistente. Quando uma estação tem dados a transmitir, ela primeiro escuta o canal para ver se mais alguém está transmitindo no momento. Se o canal estiver ocupado, a estação esperará até que ele fique ocioso. Quando detectar um canal desocupado, a estação transmitirá um quadro. Se ocorrer uma colisão, a estação esperará um intervalo de tempo aleatório e começará tudo de novo. Esse protocolo é denominado 1-persistente, porque a estação transmite com probabilidade 1 sempre que encontra o canal desocupado.

O retardo de propagação tem um efeito importante sobre o desempenho do protocolo. Há poucas chances de, logo após uma estação começar a transmitir, outra estação fique pronta para transmitir e escutar o canal. Se o sinal da primeira estação ainda não tiver atingido à segunda, esta detectará um canal desocupado e também começará a transmitir, resultando em uma colisão. Quanto maior for o retardo de propagação, maior será a importância desse efeito e pior será o desempenho do protocolo. Mesmo que o retardo de propagação seja zero, ainda assim haverá colisões. Se duas estações ficarem prontas durante a transmissão de uma terceira, ambas terão de esperar educamente até que a transmissão se encerre, e depois as duas começarão a transmitir ao mesmo tempo, resultando em uma colisão. Se elas não fossem tão impacientes, haveria menos colisões. Mesmo assim, esse protocolo é bem melhor que o ALOHA puro, pois ambas as estações respeitam a transmissão e desistem de interferir em um quadro de uma terceira estação. Intuitivamente, esse procedimento leva a um desempenho superior ao do ALOHA.

CSMA não persistente: Um segundo protocolo com detecção de portadora é o CSMA não persistente. Nesse protocolo, é feita uma tentativa consciente de ser menos ávido que no protocolo anterior. Antes de transmitir, uma estação escuta o canal. Se ninguém mais estiver transmitindo, a estação iniciará a transmissão. No entanto, se o canal já estiver sendo utilizado, a estação não permanecerá escutando continuamente a fim de se apoderar de imediato do canal após detectar o fim da transmissão anterior. Em vez disso, a estação aguardará durante um intervalo de tempo aleatório e, em seguida, repetirá o algoritmo. Conseqüentemente, esse

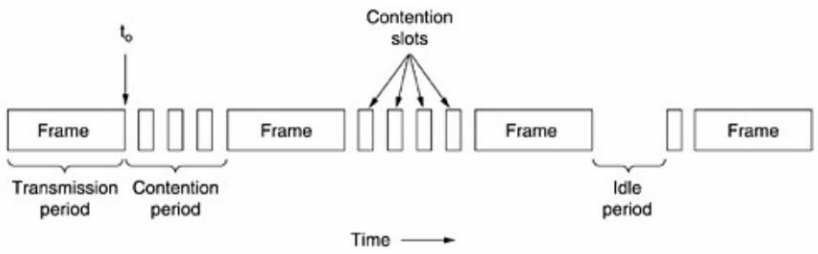
algoritmo leva a uma melhor utilização do canal, e a retardos maiores do que no CSMA 1-persistente.

O último protocolo é o CSMA p-persistente. Ele se aplica a canais segmentados (*slotted channels*) e funciona da forma apresentada a seguir. Quando está pronta para transmitir, a estação escuta o canal. Se ele estiver desocupado, a estação transmitirá com uma probabilidade p. Com uma probabilidade q = 1 - p, haverá um adiamento até o próximo *slot*. Se esse *slot* também estiver desocupado, haverá uma transmissão ou um novo adiamento, com probabilidades p e q. Esse processo se repete até o quadro ser transmitido ou até que outra estação tenha iniciado uma transmissão. Nesse último caso, ela age como se tivesse ocorrido uma colisão (ou seja, aguarda durante um intervalo aleatório e reinicia a transmissão). Se inicialmente detectar que o canal está ocupado, a estação esperará pelo próximo *slot* e aplicará o algoritmo anterior. A Figura 4.4 mostra o *throughput* calculado em comparação com o tráfego oferecido para todos os três protocolos, bem como para o ALOHA puro e o *slotted* ALOHA. Figura 4.4: Comparação entre a utilização do canal e a carga de vários protocolos de acesso Aleatório.



1.3.5 CSMA com detecção de colisões: Os protocolos CSMA persistentes e não persistentes são claramente um avanço em relação ao ALOHA, pois garantem que nenhuma estação começará a transmitir quando perceber que o canal está ocupado. Outro avanço consiste no fato de as estações cancelarem suas transmissões logo que detectam uma colisão. Em outras palavras, se duas estações perceberem que o canal está desocupado e começarem a transmitir simultaneamente, ambas detectarão a colisão quase de imediato. Em vez de terminar de transmitir seus quadros que de qualquer forma já estarão irremediavelmente adulterados, elas devem interromper a transmissão de forma abrupta tão logo a colisão for detectada. A interrupção rápida dos quadros com erros economiza tempo e largura de banda. Esse protocolo, conhecido como CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*), é amplamente usado na sub-camada MAC de LANs. Em particular, ele é a base da conhecida LAN *Ethernet*;

Figura 4.5: O CSMA/CD pode estar em um destes três estados: disputa, transmissão ou inatividade



O CSMA/CD e vários outros protocolos de LANs utilizam o modelo conceitual apresentado na Figura 4.5. No ponto marcado com t0, uma estação terminou a transmissão de um quadro. Qualquer outra estação que tenha um quadro a ser enviado pode transmiti-lo. Se duas ou mais estações decidirem transmitir simultaneamente, haverá uma colisão. As colisões podem ser detectadas verificando-se a potência e a largura do pulso do sinal recebido e comparando-o com o sinal transmitido.
Após detectar uma colisão, uma estação cancela sua transmissão, espera um intervalo de tempo aleatório e, em seguida, tenta novamente, supondo que nenhuma outra estação tenha começado a transmitir nesse ínterim. Dessa forma, o nosso modelo de CSMA/CD consistirá em períodos alternados de disputa e de transmissão, com a ocorrência de períodos de inatividade quando todas as estações estiverem em repouso (por exemplo, por falta de trabalho).

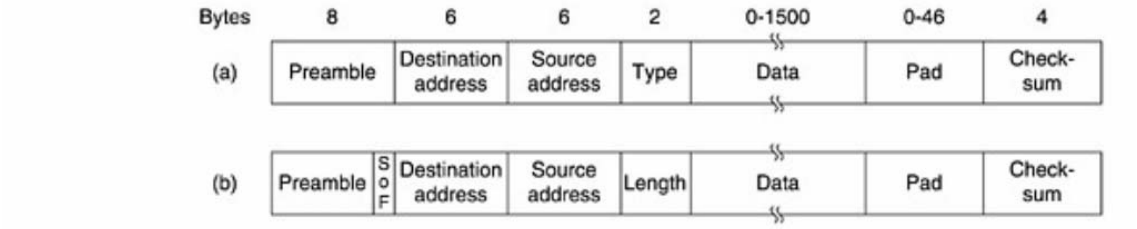
Resumo:

Figura 4.52: Métodos e sistemas de alocação de canais para um canal comum

Método	Descrição
FDM	Dedica uma banda de freqüência a cada estação
WDM	Um esquema de FDM dinâmico para fibra
TDM	Dedica um slot de tempo a cada estação
ALOHA	puro Transmissão não sincronizada em qualquer momento
Slotted ALOHA	Transmissão aleatória em slots de tempo bem definidos
CSMA 1-persistente	CSMA padrão
CSMA não persistente	Retardo aleatório quando o canal é detectado como ocupado
CSMA P-persistente	CSMA, mas com a probabilidade p de persistência
CSMA/CD	CSMA, mas cancela a operação ao detectar uma colisão
Ethernet	CSMA/CD com recuo binário exponencial

2. O Quadro ETHERNET:

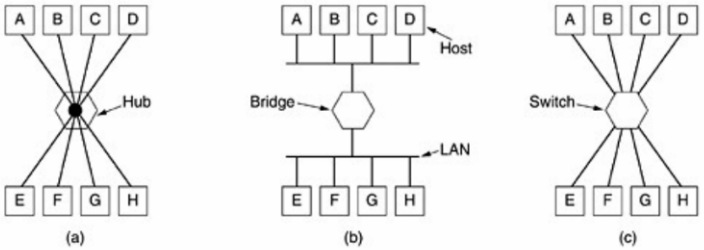
Figura 4.17: [FL] Formatos de quadros. (a) DIX Ethernet. (b) IEEE 802.3



3. Switchs, Pontes e Hubs:

Agora, vamos passar à camada de enlace de dados, onde encontramos pontes e switches. Acabamos de estudar as pontes com certa profundidade. Uma ponte conecta duas ou mais LANs, como mostra a Figura 4.47(b). Quando um quadro chega, o software da ponte extrai o endereço de destino do cabeçalho de quadro e examina uma tabela, com a finalidade de verificar para onde deve enviar o quadro. No caso de uma rede Ethernet, esse endereço é o destino de 48 bits mostrado na Figura 4.17. Como um hub, uma ponte moderna tem placas de linha, em geral para quatro ou oito linhas de entrada de um certo tipo. Uma placa de linha para *Ethernet* não pode lidar, digamos, com quadros *token ring*, porque não sabe onde encontrar o endereço de destino no cabeçalho do quadro. Porém, uma ponte pode ter placas de linha para diferentes tipos de redes e diferentes velocidades. Com uma ponte, cada linha é seu próprio domínio de colisão, em contraste com um hub.

Figura 4.47: (a) Um hub. (b) Uma ponte. (c) Um switch



Os *switches* são semelhantes a pontes pelo fato de ambos basearem o roteamento em endereços de quadro. Na verdade, muitas pessoas utilizam os dois termos de forma intercambiável. A principal diferença é que um switch é usado com maior freqüência para conectar computadores individuais, como mostra a Figura 4.47(c). Como conseqüência, quando o host A da Figura 4.47(b) quer enviar um quadro para o host B, a ponte recebe o quadro, mas simplesmente o descarta. Em contraste, na Figura 4.47(c), o *switch* deve encaminhar ativamente o quadro de A até B, porque não há outro caminho que o quadro possa seguir.

Até o momento, vimos repetidores e hubs, que são bastante semelhantes, bem como pontes e switches, que também são bem parecidos. Agora vamos passar para os roteadores, diferentes de todos os dispositivos anteriores. Quando um pacote entra em um roteador, o cabeçalho de quadro e o final são retirados, e o pacote localizado no campo de carga útil do quadro é repassado ao software de roteamento. Esse software utiliza o cabeçalho de pacote para escolher uma linha de saída. No caso de um pacote IP, o cabeçalho do pacote conterá um endereço de 32 bits (IPv4) ou de 128 bits (IPv6), mas não um endereço 802 de 48 *bits*. O *software* de roteamento não vê os endereços de quadro e nem mesmo sabe se o pacote veio de uma LAN ou de uma linha ponto a ponto. Estudaremos os roteadores e o roteamento no Capítulo 5 do *Tanenbaum*.

Parte 6: A Camada de Rede

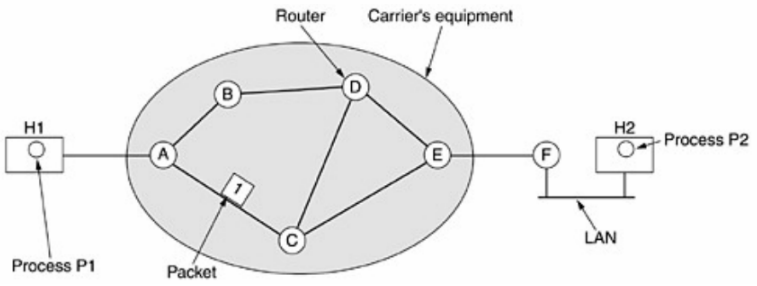
1. A Camada de Rede

A camada de rede está relacionada à transferência de pacotes da origem para o destino. Chegar ao destino pode exigir vários hops (saltos) em roteadores intermediários ao longo do percurso. Essa função contrasta claramente com a função da camada de enlace de dados, que tem o objetivo mais modesto de apenas mover quadros de uma extremidade de um fio até a outra. Portanto, a camada de rede é a camada mais baixa que lida com a transmissão fim a fim. Para atingir seus objetivos, a camada de rede deve conhecer a topologia da sub-rede de comunicações (ou seja, o conjunto de todos os roteadores) e escolher os caminhos mais apropriados através dela. A camada de rede também deve ter o cuidado de escolher rotas que evitem sobrecarregar algumas das linhas de comunicação e roteadores enquanto deixam outras ociosas. Por fim, quando a origem e o destino estão em redes diferentes, ocorrem novos problemas, e cabe à camada de rede lidar com eles. Neste capítulo, estudaremos todas essas questões e as ilustraremos, quando principalmente a Internet e o protocolo de sua camada de rede, o IP, embora também sejam examinadas as redes sem fios.

2. Questões de Projetos da camada de rede:

2.1 Comutação de pacotes store-and-forward: Antes de começarmos a explicar os detalhes da camada de rede, vale a pena redefinir o contexto em que operam os protocolos a camada de rede. Esse contexto pode ser visto na Figura 5.1. Os principais componentes do sistema são o equipamento da concessionária de comunicações (roteadores conectados por linhas de transmissão), mostrados na elipse sombreada, e o equipamento dos clientes, mostrado fora da elipse. O host H1 está diretamente conectado a um dos roteadores da concessionária de comunicações, denominado A, por uma linha dedicada. Em contraste, H2 está em uma LAN com um roteador F pertencente ao cliente e operado por ele. Esse roteador também tem uma linha dedicada para o equipamento da concessionária de comunicações. Mostramos F fora da elipse porque ele não pertence à concessionária de comunicações; porém, em termos de construção, software e protocolos, é bem provável que ele não seja diferente dos roteadores da concessionária de comunicações. O fato de ele pertencer à sub-rede é discutível mas, para os propósitos deste capítulo, roteadores no local do cliente são considerados parte da sub-rede, porque escutam os mesmos algoritmos que os roteadores da concessionária de comunicações (e nossa principal preocupação aqui é o estudo dos algoritmos).

Figura 5.1: O ambiente dos protocolos da camada de rede



2.2 Serviços oferecidos à camada de transporte

A camada de rede oferece serviços à camada de transporte na interface entre a camada de rede e a camada de transporte. Uma questão importante é identificar os tipos de serviços que a camada de rede oferece à camada de transporte. Os serviços da camada de rede foram projetados tendo em vista os objetivos a seguir.

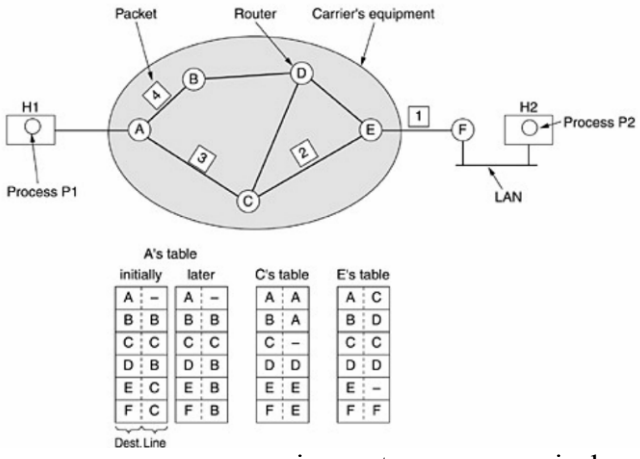
1. Os serviços devem ser independentes da tecnologia de roteadores.
2. A camada de transporte deve ser isolada do número,do tipo e da topologia dos roteadores presentes.
3. Os endereços de rede que se tornaram disponíveis para a camada de transporte devem usar um plano de numeração uniforme, mesmo nas LANs e WANs.

Tendo definido esses objetivos, os projetistas da camada de rede têm muita liberdade para escrever especificações detalhadas dos serviços a serem oferecidos à camada de transporte. Essa liberdade costuma se transformar em uma violenta batalha entre duas facções. A discussão se concentra na seguinte questão: a camada de rede deve fornecer serviço orientado a conexões ou serviço sem conexões?

2.3 Implementação do serviço sem conexões

Depois de analisar as duas classes de serviço que a camada de rede pode oferecer a seus usuários, chegou a hora de vermos como essa camada funciona por dentro. São possíveis duas organizações diferentes, dependendo do tipo de serviço oferecido. Se for oferecido o serviço sem conexões, os pacotes serão injetados individualmente na sub-rede e roteados de modo independente uns dos outros. Não será necessária nenhuma configuração antecipada. Nesse contexto, os pacotes freqüentemente

são chamadas datagramas (em uma analogia com os telegramas) e a sub-rede será denominada sub-rede de datagramas. Se for usado o serviço orientado a conexões, terá de ser estabelecido um caminho desde o roteador de origem até o roteador de destino, antes de ser possível enviar quaisquer pacotes de dados. Essa conexão é chamada circuito virtual, em analogia com os circuitos físicos estabelecidos pelo sistema telefônico, e a sub-rede é denominada sub-rede de circuitos vituais. Nesta seção, examinaremos as sub-redes de datagramas; na próxima, estudaremos as sub-redes de circuitos virtuais. Vejamos agora como funciona uma sub-rede de datagramas. Suponha que o processo P1 da Figura 5.2 tenha uma longa mensagem para P2. Ele entrega a mensagem à camada de transporte, com instruções para que ela seja entregue a P2 do host H2. O código da camada de transporte funciona em H1, em geral dentro do sistema operacional. Ele acrescenta um cabeçalho de transporte ao início da mensagem e entrega o resultado à camada de rede, que talvez seja simplesmente outro procedimento no sistema operacional. A figura abaixo exibe o roteamento em uma sub-rede de datagramas.



Vamos supor que a mensagem seja quatro vezes mais longa que o tamanho máximo de pacote, e portanto que a camada de rede tem de dividi-la em quatro pacotes, 1, 2, 3 e 4, e enviar cada um deles ao roteador A, usando algum protocolo ponto a ponto como, por exemplo, o PPP . Nesse ponto, a concessionária de comunicações assume o controle. T odo roteador tem uma tabela interna que informa para onde devem ser enviados os pacotes a serem entregues a cada destino possível. Cada entrada da tabela é um par que consiste em um destino e na linha de saída a ser utilizada para esse destino. Somente podem ser usadas linhas conectadas direta mente. Por exemplo, na figura acima, em A temos apenas duas linhas de saída — para D e C — e assim todo pacote recebido deve ser enviado a um desses roteadores, mesmo que o destino final seja algum outro roteador. A tabela de roteamento inicial de A é mostrada na figura sob o título "Inicialmente".

À medida que chegaram ao roteador A, os pacotes 1, 2 e 3 foram armazenados por algum tempo (para que seus totais de verificação fossem conferidos). Em seguida, cada um deles foi encaminhado para C, de acordo com a tabela de A. O pacote 1 foi então encaminhado para E e depois para F . Chegando a F, ele foi encapsulado em um quadro da camada de enlace de dados e transmitido para H2 pela LAN. Os pacotes 2 e 3 seguiram a mesma rota. Entretanto, aconteceu algo diferente com o pacote 4. Quando chegou ao roteador A, ele foi enviado para o roteador B, embora seu destino também fosse F . Por alguma razão, A decidiu enviar o pacote 4 por uma rota diferente da que foi usada para os três primeiros pacotes. T alvez ele tenha tomado conhecimento de uma obstrução de tráfego em algum lugar no caminho ACE e tenha atualizado sua tabela de roteamento, como mostamos na figura sob o título

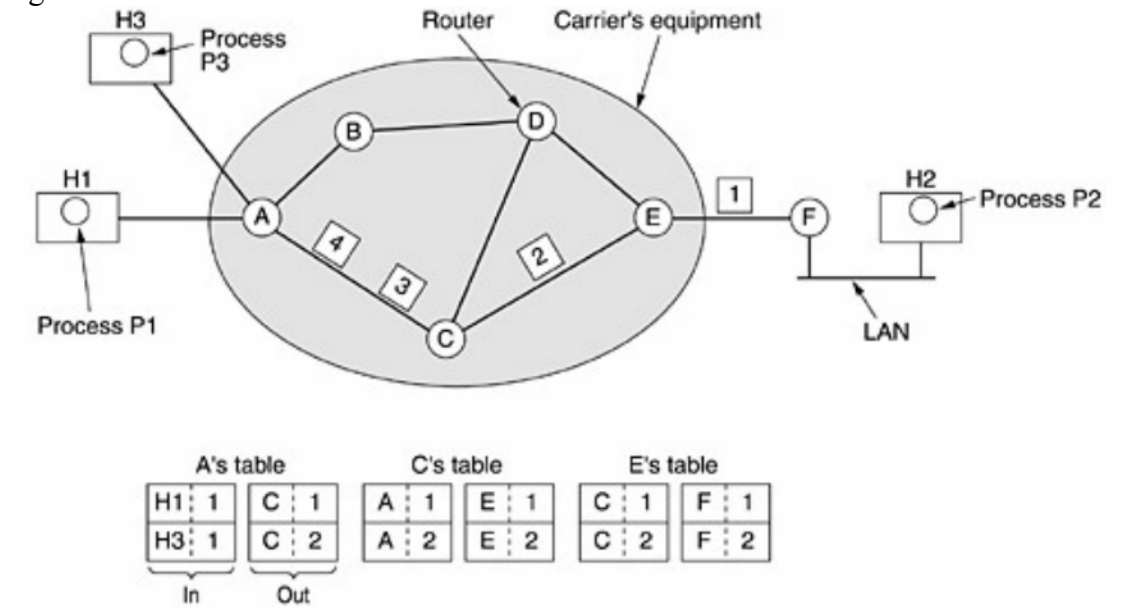
"Mais tarde". O algoritmo que gerencia as tabelas e toma as decisões de roteamento é chamado algoritmo de roteamento. Os algoritmos de roteamento constituem um dos principais assuntos que estudaremos neste capítulo.

2.4 Implementação do Serviço Orientado a Conexões: A No caso do serviço orientado a conexões, precisamos de uma sub-rede de circuitos virtuais. Vejamos como ela funciona. A idéia que rege os circuitos virtuais é evitar a necessidade de escolher uma nova rota para cada pacote enviado, como na Figura anterior. Em vez disso, quando uma conexão é estabelecida, escolhe-se uma rota desde a máquina de origem até a máquina de destino, como parte da configuração da conexão, e essa rota é armazenada em tabelas internas dos roteadores. A rota é usada por todo o tráfego que flui pela conexão, exatamente como ocorre no sistema telefônico. Quando a conexão é liberada, o circuito virtual também é encerrado. Com o serviço orientado a conexões, cada pacote transporta um identificador, informando a que circuito virtual ele pertence.

Como exemplo, considere a situação da Figura abaixo. Na figura, o host H1 estabeleceu a conexão 1 com o host H2. Ela é memorizada como a primeira entrada de cada uma das tabelas de roteamento. A primeira linha da tabela de A informa que, se um pacote contendo o identificador de conexão 1 chegar de H1, ele será enviado ao roteador C e receberá o identificador de conexão 1. De modo semelhante, a primeira entrada em C faz o roteamento do pacote para E, também com o identificador de conexão 1.

Agora, vamos considerar o que acontece se H3 também quiser estabelecer uma conexão para H2. Ele escolhe o identificador de conexão 1 (porque está iniciando a conexão, e essa é sua única conexão) e informa à sub-rede que ela deve estabelecer o circuito virtual. Isso conduz à segunda linha nas tabelas. Observe que nesse caso temos um conflito porque, embora A possa distinguir facilmente os pacotes da conexão 1 provenientes de H1 dos pacotes da conexão 1 que vêm de H3, C não tem como fazer o mesmo. Por essa razão, A atribui um identificador de conexão diferente ao tráfego de saída correspondente a segunda conexão. Evitar conflitos desse tipo é a razão pela qual os roteadores precisam ter a capacidade de substituir identificadores de conexões em pacotes de saída. Em alguns contextos, essa operação é chamada troca de rótulos.

Figura 5.3: Roteamento em uma sub-rede de circuitos virtuais:



3. Comparação Entre Sub-Redes De Circuitos Virtuais E De Datagramas:

Dentro da sub-rede, existem vários compromissos entre circuitos virtuais e datagramas, como o compromisso entre espaço de memória do roteador e largura de banda. Os circuitos virtuais permitem que os pacotes contenham números de circuitos em vez de endereços de destino completos. O preço pago pelo uso de circuitos virtuais é o espaço na tabela dentro dos roteadores.

Outro compromisso é o que se dá entre o tempo de configuração e o tempo de análise e endereço. O uso de circuitos virtuais requer uma fase de configuração que leva tempo e consome recursos. Entretanto, é fácil descobrir o que fazer com um pacote de dados em uma sub-rede de circuitos virtuais. O roteador só utiliza o número do circuito para criar um índice em uma tabela e descobrir para onde vai o pacote.

Em uma sub-rede de datagramas, é necessário um procedimento de pesquisas mais complicado para localizar a entrada correspondente ao destino.

Os circuitos virtuais têm vantagens na garantia da qualidade de serviço (QoS) e ao evitar o congestionamento dentro de uma sub-rede, pois os recursos podem ser reservados antecipadamente, o que não ocorre na rede de datagramas.

Os circuitos virtuais têm um problema grande com a vulnerabilidade. Se um roteador apresentar uma falha e perder sua memória, mesmo que volte um segundo depois, todos os circuitos virtuais que estiverem passando por ele terão de ser interrompidos. No roteador de datagramas somente os usuários cujos pacotes estiverem enfileirados no roteador naquele momento serão afetados. A perda de uma linha de comunicação é fatal para os circuitos virtuais, mas podem ser compensadas se utilizados datagramas, pois estes permitem que os roteadores equilibrem o tráfego pela sub-rede uma vez que a rota pode ser parcialmente alterada.

Figura 5.4: Comparação entre sub-redes de circuitos virtuais e de datagramas

Questão	Sub-rede de datagramas	Sub-rede de circuitos virtuais
Configuração de circuitos	Desnecessária	Obrigatória
Endereçamento	Cada pacote contém os endereços de origem e de destino completos	Cada pacote contém um número de circuito virtual curto
Informações sobre o estado	Os roteadores não armazenam informações sobre o estado das conexões	Cada circuito virtual requer espaço em tabelas de roteadores por conexão
Roteamento	Cada pacote é roteado independentemente	A rota é escolhida quando o circuito virtual é estabelecido; todos os pacotes seguem essa rota
Efeito de falhas no roteador	Nenhum, com exceção dos pacotes perdidos durante a falha	Todos os circuitos virtuais que tiverem passado pelo roteador que apresentou o defeito serão encerrados
Qualidade de serviço	Difícil	Fácil, se for possível alocar recursos suficientes com antecedência para cada circuito virtual
Controle de congestionamento	Difícil	Fácil, se for possível alocar recursos suficientes com antecedência para cada circuito virtual

4. Algoritmos de roteamento

A principal função da camada de rede é rotear pacotes da máquina de origem para a máquina de destino. Na maioria das sub-redes, os pacotes necessitarão de vários *hops* para cumprir o trajeto. A única exceção importante diz respeito às redes de difusão, mas mesmo aqui o roteamento depende do fato de a origem e o destino não estarem na mesma rede. Os algoritmos que escolhem as rotas e as estruturas de dados que eles utilizam constituem um dos elementos mais importantes do projeto da camada de rede.

O algoritmo de roteamento é a parte do software da camada de rede responsável pela decisão sobre a linha de saída a ser usada na transmissão do pacote de entrada. Se a sub-rede utilizar datagramas internamente, essa decisão deverá ser tomada mais uma vez para cada pacote de dados recebido, pois a melhor rota pode ter sido alterada desde a última vez. Se a sub-rede utilizar circuitos virtuais internamente, as decisões de roteamento serão tomadas somente quando um novo circuito virtual estiver sendo estabelecido. Daí em diante, os pacotes de dados seguirão a rota previamente estabelecida. Às vezes, essa última circunstância é chamada roteamento por sessão, pois uma rota permanece em vigor durante toda uma sessão do usuário (por exemplo, uma sessão de *login* em um terminal ou uma transferência de arquivos).

Algumas vezes, é útil fazer distinção entre roteamento, que é a tomada de decisão sobre quais rotas utilizar, e encaminhamento, o que acontece quando um pacote chega. Podemos imaginar que um roteador tem dois processos em seu interior. Um deles trata cada pacote que chega, procurando a linha de saída que será usada para ele nas tabelas de roteamento. Esse processo é o encaminhamento.

O outro processo é responsável pelo preenchimento e pela atualização das tabelas de roteamento. É nesse processo que o algoritmo de roteamento entra em cena. Mesmo que as rotas sejam escolhidas independentemente para cada pacote ou apenas quando novas conexões são estabelecidas, certas propriedades são desejáveis em um algoritmo de roteamento: correção, simplicidade, robustez, estabilidade, eqüidade e otimização.

Os itens correção e simplicidade são auto explicativos mas, em princípio, talvez a necessidade de robustez seja menos óbvia. Uma vez que uma rede de maior porte é instalada, espera-se que ela funcione continuamente durante anos sem apresentar qualquer falha no sistema. Durante esse período, haverá falhas de hardware e software de todos os tipos. Os hosts, os roteadores e as linhas irão falhar repetidamente, e a topologia mudará muitas vezes.

O algoritmo de roteamento deve ser capaz de aceitar as alterações na topologia e no tráfego sem exigir que todas as tarefas de todos os *hosts* sejam interrompidas e que a rede seja reinicializada sempre que algum roteador apresentar falha.

Quando alguns cientistas que fundaram a Cisco perceberam que poderiam utilizar filtragem da camada três do modelo de referência OSI para melhorar o desempenho da rede, eles desenvolveram o roteador.

Para facilitar o conceito imagine a situação a seguir. Quando você sai de São Paulo e vai para as praias da o Rio de Janeiro de carro e, como bom motorista, usa o mapa para chegar lá. Note que existem dois caminhos: Por qual deles ir? Pelo caminho mais curto seria a resposta mais óbvia. No entanto, seria mais sensato responder que depende. Apesar de ser tentador ir pelo caminho mais curto, é necessário considerar alguns pontos:

1 **Tráfego:** Qual dos dois caminhos tem um tráfego menor?
2 **Estado de conservação**: Qual dos dois caminhos tem menos buracos.
3 **Distância**: Pode ser considera sim. Observe o próximo item.
4 **Análise dos pontos anteriores:** Apesar de buracos e do tráfego ou mesmo a distância maior, pode ser interessante ir naquele caminho do que o mais curto.

Assim trabalha o roteador. Ele, através de análise do *link* de comunicação, permite uma comunicação entre redes, pois trabalha na camada de rede, ou seja, lida diretamente com o *Internet Protocol*.

O algoritmo de roteamento deve ser capaz de aceitar as alterações na topologia e no tráfego sem exigir que todas as tarefas de todos os hosts sejam interrompidas e que a

rede seja reinicializada sempre que algum roteador apresentar falha. Tais algoritmos podem ser agrupados em duas classes:

- **Adaptativos; e**
- **Não-Adaptativo**s.

Os não-adaptativos não baseiam suas decisões de roteamento em medidas ou estimativas do tráfego e da topologia atuais. Este tipo de roteamento é denominado roteamento estático, uma vez que a escolha da rota é definida previamente off-line. Os algoritmos adaptativos mudam suas decisões de roteamento para refletir mudanças na topologia e, normalmente, também no tráfego. Estes diferem em termos do lugar em que obtêm suas informações, seja no próprio local, em roteadores adjacentes ou todos os roteadores.

4.1 Roteamento pelo caminho mais curto

Esta é uma técnica muito utilizada, haja vista ser simples e fácil de entender. A idéia principal é criar um grafo de sub-rede, com cada nó do grafo representando um roteador e cada arco indicando uma linha de comunicação (enlace). Para escolher uma rota, o algoritmo simplesmente encontra o caminho mais curto expresso na rota. A métrica usada para determinar o caminho mais curto entre fonte e destino pode se basear em diferentes métodos:

- Número de *hops* (saltos) entre fonte e destino;
- Distância Física (Geográfica);
- Fila média e atraso de transmissão associados a cada arco no caminho para algum pacote padrão de teste transmitido a intervalos regulares. O caminho mais curto é o mais rápido, ao invés daquele com menor número de arcos ou km.

Em geral os *labels* nos arcos podem ser computados como função de mais de um argumento, entre os quais:

- Distância;
- Custo de comunicação;
- Largura de banda;
- Tamanho médio da Fila;
- Tráfego Médio;
- *Delays* (atrasos).

O algoritmo pode calcular o caminho mais curto através de um dos critérios citados ou através de uma combinação ponderada dos diferentes critérios.

4.2 Flooding (Inundação): O algoritmo de inundação é um algoritmo estático no qual cada pacote de entrada é enviado para todas as linhas de saída, exceto para aquela em que chegou. É gerada uma vasta quantidade de pacotes duplicados, a menos que algumas medidas sejam tomadas para tornar mais lento o processo. Uma dessas medidas é ter um contador de *hops* contido no cabeçalho de cada pacote. O contador é decrementado até atingir zero. Normalmente ele tem o número que saltos necessários para percorrer todo o caminho. Outro meio é contar quais pacotes foram transmitidos para que eles não sejam enviados novamente. É usado para distribuir informação para todos os nós que age como um receptor e transmissor de mensagem.

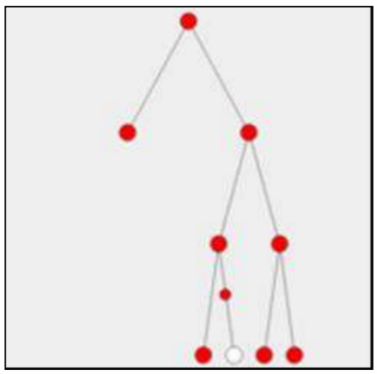


Figura 5 – Inundação

4.3 Roteamento por difusão: Em algumas aplicações, os *hosts* precisam enviar mensagem a muitos outros hosts. Neste caso o envio de um pacote a todos os destinos simultaneamente é chamado de difusão (*broadcasting*). O método exige que a origem tenha uma lista completa de todos os destinos. Na prática, essa pode ser a única possibilidade completa de todos os destino receberem o pacote transmitido.

4.4 Roteamento para *hosts* móveis: Como visto, a principal função da camada de rede é rotear pacotes. Na maioria das sub-redes os pacotes necessitarão de vários saltos para cumprir seu trajeto. Para isso são implementados diversos tipos de algoritmos de roteamento, dentre eles o roteamento para *hosts* móveis. É comum hoje em dia pessoas utilizarem computadores pessoais para ler mensagens de *e-mail*, acessar *internet*, etc. Esses *hosts* móveis criam uma nova complicação: antes de rotear um pacote para um *host* móvel, primeiro a rede precisa localizá-lo. Para resolver essa problemática definiu-se para cada área um ou mais agentes externos, que controlam todos os usuários móveis que visitam a área. Além disso, cada área possui um agente interno, que controla os usuários cujas bases estejam na área, mas que estejam no momento visitando outra área. Quando um novo usuário entra em uma área, conectando-se a ela (por exemplo, ligando seu computador na LAN), ou simplesmente percorrendo a célula, seu computador deve-se registrar com o agente externo dessa área. Periodicamente, cada agente externo transmite um pacote anunciando sua existência e endereço. Um *host* móvel recém-chegado pode aguardar uma dessas mensagens; no entanto, se nenhuma chegar rápido o suficiente, o host móvel poderá transmitir um pacote dizendo: "Existe algum agente externo?”

- O *host* móvel é registrado com o agente externo, fornecendo seu endereço fixo, o endereço atual de camada de enlace de dados e algumas informações de segurança.
- O agente externo contacta o agente interno do host móvel e diz: "Um de seus hosts está aqui".
- A mensagem do agente externo para o agente interno contém o endereço de rede do agente externo.
- A mensagem contém ainda as informações de segurança, para convencer o agente interno de que o *host* móvel está realmente lá.
- O agente interno examina as informações de segurança, que contêm um timbre de hora, para provar que foi gerando há alguns segundos. Se estiver tudo de acordo, o agente interno diz ao externo para prosseguir.
- Quando o agente externo obtém a confirmação do agente interno, ele cria uma entrada em sua tabela e informa ao *host* móvel que agora ele está registrado.

- O ideal é que quando o usuário sair de uma área, isso também seja divulgado para permitir o cancelamento do registro, mas muitos usuários desligam seus computadores abruptamente quando terminam de usá-los.

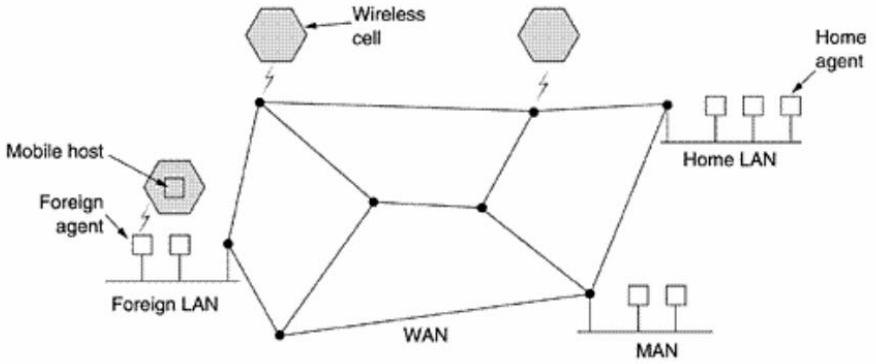


Figura 6 - Roteamento para hosts móveis

Quando é enviado a um usuário móvel, o pacote é roteado para a LAN básica do usuário, pois é isso que o endereço diz que deve ser feito. Os pacotes enviados para o usuário móvel através de sua LAN básica são interceptados pelo agente interno. Em seguida, o agente interno consulta a nova localização (temporária) do usuário móvel e encontra o endereço do agente externo que está tratando do usuário móvel.

5. Algoritmos De Controle De Congestionamento

Congestionamento ocorre quando a quantidade de pacotes na rede é muito grande (normalmente isso ocorre quando se atinge um patamar da capacidade de carga dos canais de comunicação).

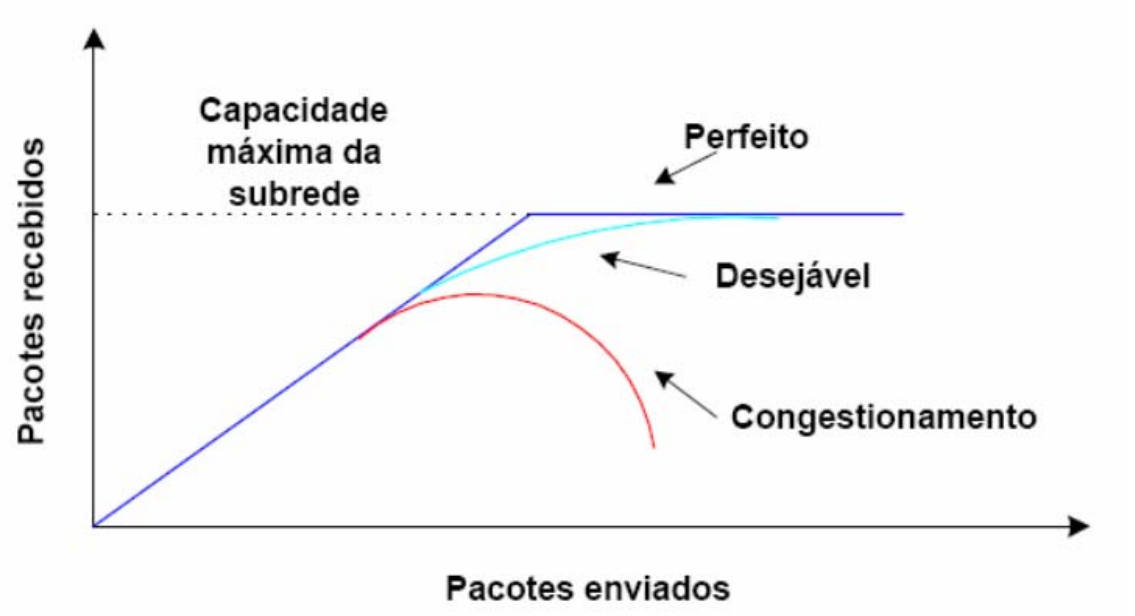


Figura 7 - Janela de congestionamento

As causas do congestionamento podem ser as seguintes:

- Pacotes chegando por canais de comunicação rápidos, tendo de sair por canais mais lentos;
- Roteadores lentos;

- Roteadores com pouca memória para armazenar pacotes temporariamente;

Contudo, existem dois métodos que podem ser usados para controlar o fluxo, a saber:

- Modelo circuito aberto; e
- Modelo circuito fechado.

O primeiro propõe resolver os problemas na fase de projeto/configuração dos roteadores de modo a (tentar) garantir que não ocorra congestionamento. Para ajustar alguma coisa, é necessário de reinicializar tudo.

O segundo realiza a monitoração do sistema para detectar quando e onde o congestionamento ocorre. Realiza a passagem dessas informações para pontos de controle onde alguma ação pode ser tomada e realiza o ajuste da operação do sistema para corrigir o problema. Neste modelo de circuito fechado, o controle pode ser explícito, quando o ponto de congestionamento avisa (de alguma forma) a origem dos pacotes (exemplo: ATM); ou implícito, quando a origem dos pacotes deduz que há congestionamento fazendo observações localmente (exemplo: pela demora no recebimento de confirmação de entrega de pacotes - TCP/IP).

5.1 Políticas de prevenção de congestionamento: As políticas adotadas em diversas camadas de uma pilha de protocolos que afetam o congestionamento são expressas na tabela abaixo:

Tabela 1 - Políticas de controle de congestionamento

CAMADA	POLÍTICAS
Transporte	- Política de retransmissão; - Política de armazenamento para segmentos fora de ordem; - Política de reconhecimento (Ack) de segmento; - Política de controle de fluxo; - Determinação do temporizador (*timeout*);
Rede	- Modo circuito virtual versus modo datagrama; - Enfileiramento de pacote e política de serviço; - Política de descarte de pacote; - Algoritmo de roteamento; - Gerência de tempo de vida de pacote;
Enlace de dados	- Política de retransmissão; - Política de armazenamento para quadro fora de ordem (*go back* N, *seletive repeat*); - Política de reconhecimento (Ack) de quadro (com/sem *pigybacking*); - Política de controle de fluxo (com/sem *slide window*).

5.2 Controle de congestionamento em sub-redes de circuitos virtuais: A maioria dos algoritmos de congestionamento tenta impedir que o congestionamento ocorra, em vez de lidar com ele após seu surgimento. Uma técnica muito utilizada para impedir que um congestionamento iniciado se torne pior é o **Controle de Admissão.** Uma vez que o congestionamento tenha dado alguma indicação de sua existência, nenhum outro circuito virtual será estabelecido até que o problema tenha passado. Portanto, todas as tentativas de estabelecer novas conexões da camada de transporte falharão. Alternativa é permitir novos circuitos, mas rotear com cuidado todos os novos circuitos virtuais em áreas problemáticas.

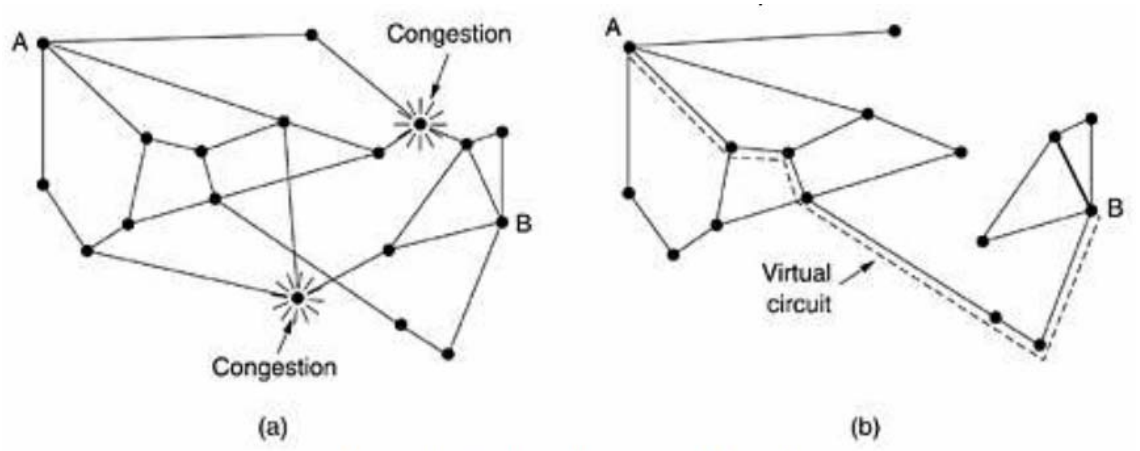


Figura 8 - Sub-rede congestionada

A idéia neste caso, como se vê na figura acima é redesenhar a sub-rede (b), omitindo os roteadores congestionados e todas as suas linhas. Existe ainda outra situação que é um acordo entre o host e a sub-rede, na qual fica acordado, no momento da configuração do circuito virtual, o volume, a formatação do tráfego, a qualidade de serviço exigida e outros parâmetros. Para isso a sub-rede reservará recursos ao longo do caminho. Assim, é improvável que ocorra congestionamento nos novos circuitos virtuais. A desvantagem desse método é o desperdício de recursos.

5.3 Controle de congestionamento em sub-redes de datagramas: No caso das sub-redes de datagramas, cada roteador pode monitorar facilmente a utilização de suas linhas de saída e de outros recursos. O roteador pode associar a cada linha uma variável o qual reflete a utilização da linha. Assim, cada pacote recém-chegado é conferido para saber se sua linha de saída encontra-se em estado de advertência. Havendo algum estado de advertência, uma ação será tomada, dentre elas:

5.3.1 BIT DE ADVERTÊNCIA: Assinala o estado de advertência ativando um *bit* especial no cabeçalho do pacote, feito no *frame relay*. Quando o pacote chega ao destino, a entidade de transporte copiava o *bit* na próxima confirmação a ser enviada de volta à origem. Em seguida, a origem interrompia o tráfego. Enquanto estivesse no estado de advertência, o roteador continuava a definir o *bit* de advertência, e isso significava que a origem continuava a receber informações com o *bit* ativado. Assim, a origem monitora a fração de confirmações com o bit ativado e ajusta sua velocidade de transmissão. Enquanto os bits de advertência continuar a fluir, a origem continuava a diminuir sua taxa de transmissão. Quando diminui a velocidade de chegada das confirmações, a origem aumenta a taxa de transmissão.

5.3.2 PACOTES REGULADORES: Neste modelo o roteador enviará um pacote regulador ao host de origem, informando que deve interromper ou enviar mais lentamente os pacotes. O pacote original é marcado (um bit no cabeçalho é ativado) para que ele não venha a gerar mais pacotes reguladores ao longo do caminho e depois é encaminhado de forma habitual. Ao receber o pacote regulador, o *host* de origem é obrigado a reduzir o tráfego enviado ao destino especificado. Este método ignora novos pacotes do mesmo destino por um tempo. Isso evita o excesso de *feedback*. Contudo, não funciona bem com longas distâncias ou grande largura de banda e há certa demora para avisar a origem.

5.3.3 PACOTES REGULADORES *HOP* A *HOP*: Em altas velocidades ou em longas distâncias, o envio de um pacote regulador para os hosts de origem não

funciona bem devido à reação ser muito lenta. A solução é fazer com que o pacote regular tenha efeito em cada *hop* pelo qual passar. O efeito desse esquema é oferecer alívio rápido no ponto de congestionamento, ao preço de aumentar o consumo de *buffers* do fluxo ascendente. Assim o congestionamento pode ser cortado pela raiz sem perda de pacotes.

6. O Protocolo IP

O protocolo IP integra um sistema de entrega fim-a-fim. Dessa forma é um tipo de protocolo não orientados à conexão, sem controle de erros e sem reconhecimento. Isso significa que o protocolo IP não executa controle de erros sobre os dados da aplicação, controle de fluxo, seqüenciamento de dados e entrega ordenada.

O protocolo IP apresenta algumas características como às listadas abaixo:

- Utiliza um serviço de entrega denominado *Best-Effort* (Melhor esforço): Os pacotes não são descartados sumariamente, o protocolo torna-se não confiável somente quando há exaustão de recursos; Oferece ainda serviço não baseado em conexão.
- Apresenta um tamanho de datagrama variável. No caso do IPV4 seu tamanho máximo pode atingir 64 Kb;
- Realiza a fragmentação de datagramas com recombinação no destino. Isto garante suporte a redes intermediárias com MTU (*Maximum Transmission Unit*) diferentes;
- É responsável pelo roteamento de datagramas ou fragmentos;
- Provê envio e recebimento de erros por meio de um protocolo chamado ICMP.

É ao IP que compete levar a informação de um extremo ao outro na *Internet*, atravessando várias redes, potencialmente muito diferentes.

6.1 O Datagrama IP :

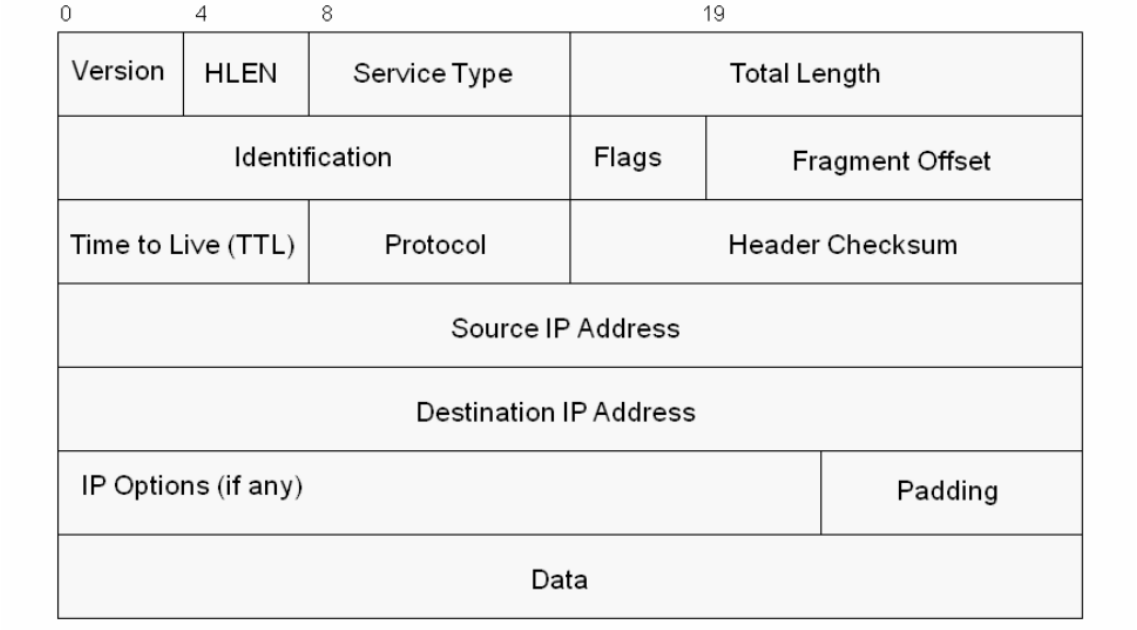


Figura 9 – Frame IP

- **Version:** A Versão do protocolo hoje é ipv4, no entanto para o futuro será trabalhado a versão ipv6.
- **Hlen:** tamanho do cabeçalho, devido a opções variáveis.
- **Service type:** Atualmente ignorado.

- Fragmentação controlada por: **Identification** (campo tem o mesmo valor para cada fragmento de um datagrama)
    - **Flags**: "don't fragment" (o destino não sabe recombinar) e "more fragments";
    - **Offset**: offset do fragmento em múltiplos de 8 bytes;
    - **Time to live**: Decrementado durante a vida do datagrama para ter certeza que tabelas de roteamento corrompidas não manterão pacotes na rede para sempre; Datagrama descartado quando TTL chega a 0. Deveriam ser segundos, mas todo mundo decrementa a cada hope.
- **Protocol:** icmp, udp, tcp, dentre outros;
- **Checksum:** controle de erro. Deve ser recalculado a cada hope devido a mudanças de ttl;
- **Source/Destination:** endereços discutidos mais à frente;
- **Padding:** Para ter cabeçalho múltiplo de 32 bits;

6.2 O Endereçamento IP

O endereço IP (*Internet Protocol*), de forma genérica, é um endereço que indica o local de um determinado equipamento (normalmente computadores) em uma rede privada ou pública. O endereço IP, na versão 4 (IPv4), é um número de 32 *bits* escrito com quatro octetos representados no formato decimal (exemplo: 128.6.4.7). A primeira parte do endereço identifica uma rede específica na inter-rede, a segunda parte identifica um *host* dentro dessa rede. Dessa forma, os endereços IP podem ser usados tanto para nos referir a redes quanto a um host individual. Podemos também nos referir a todos os hosts de uma rede através de um endereço por difusão, quando, por convenção, o campo identificador de *host* deve ter todos os *bits* iguais a 1 (um). Um endereço com todos os 32 bits iguais a 1 é considerado um endereço por difusão para a rede do host origem do datagrama.

6.2.1 CLASSES IP: O IP utiliza cinco classes diferentes de endereços, contudo, efetivamente somente três:

6.2.2. CLASSE FAIXA DE IP Nº DE HOSTS POR REDE

- **A** 1.0.0.0 até 126.0.0.0 16 777 216
- **B** 128.0.0.0 até 191.255.0.0 65 536
- **C** 192.0.0.0 até 223.255.255.254 256
- **D** 224.0.0.0 até 239.255.255.255 Multicast
- **E** 240.0.0.0 até 247.255.255.255 Uso futuro; atualmente reservada a testes pela IETF

A definição de tipo de endereço classes de endereços deve-se ao fato do tamanho das redes que compõem a inter-rede variar muito, indo desde redes locais de computadores de pequeno porte, até redes públicas interligando milhares de hosts.

6.2.2 CLASSES ESPECIAIS

Existem classes especiais na Internet que não são consideradas públicas, não são consideradas como endereçáveis, ou seja, são reservadas, por exemplo: para a comunicação com uma rede privada ou com o computador local (localhost).

6.2.3 BLOCO DE ENDEREÇOS DESCRIÇÃO REFERÊNCIA

0.0.0.0/8 Rede corrente (só funciona como endereço de origem) RFC 1700
10.0.0.0/8 Rede Privada RFC 1918
14.0.0.0/8 Rede Pública RFC 1700
39.0.0.0/8 Reservado RFC 1797
127.0.0.0/8 Localhost RFC 3330
128.0.0.0/16 Reservado (IANA) RFC 3330

169.254.0.0/16 Zeroconf RFC 3927
172.16.0.0/12 Rede Privada RFC 1918
191.255.0.0/16 Reservado (IANA) RFC 3330
192.0.0.0/24
192.0.2.0/24 Documentação RFC 3330
192.88.99.0/24 IPv6 para IPv4 RFC 3068
192.168.0.0/16 Rede Privada RFC 1918
198.18.0.0/15 Teste de benchmark de redes RFC 2544
223.255.255.0/24 Reservado RFC 3330
224.0.0.0/4 Multicasts (antiga rede Classe D) RFC 3171
240.0.0.0/4 Reservado (antiga rede Classe E) RFC 1700

6.2.4 LOCALHOST

A faixa de IP 127.0.0.0 – 127.255.255.255 (ou 127.0.0.0/8) é reservada para a comunicação com o computador local (localhost). Qualquer pacote enviado para estes endereços ficarão no computador que os gerou e serão tratados como se fossem pacotes recebidos pela rede (*Loopback*). O endereço de loopback local (127.0.0.0/8) permite à aplicação-cliente endereçar ao servidor na mesma máquina sem saber o endereço do host, chamado de localhost. Na pilha do protocolo TCP/IP, a informação flui para a camada de rede, onde a camada do protocolo IP reencaminha de volta através da pilha.

6.2.5 REDES PRIVADAS

Dos mais de 4 bilhões de endereços disponíveis, três faixas são reservadas para redes privadas. Estas faixas não podem ser roteadas para fora da rede privada - não podem se comunicar diretamente com redes públicas. Dentro das classes A, B e C foram reservadas redes (normalizados pela RFC 1918) que são conhecidas como endereços de rede privados.

6.2.6 SUB-REDES: Uma sub-rede é uma divisão de uma rede de computadores. A divisão de uma rede grande em redes menores resulta num tráfego de rede reduzido, administração simplificada e melhor performance de rede.

Para criar sub-redes, qualquer máquina tem que ter uma máscara de sub-rede que define que parte do seu endereço IP será usado como identificador da sub-rede e como identificador do host. As sub-redes servem para:

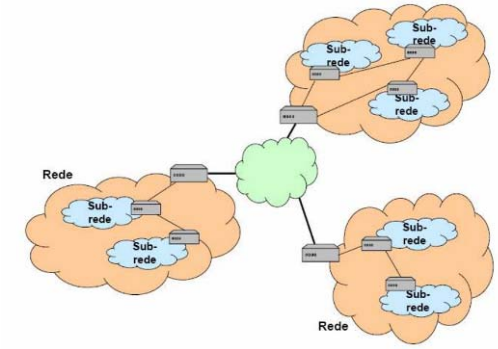
- Simplificar a administração de redes. As sub-redes podem ser usadas para delegar gestão de endereços, problemas e outras responsabilidades.
- Reconhecer a estrutura organizacional. A estrutura de uma organização (empresas, organismos públicos, etc.) pode requerer gestão de rede independente para algumas divisões da organização.
- Isolar tráfego por organização. Acessível apenas por membros da organização, relevante quando questões de segurança são levantadas.
- Isolar potenciais problemas. Se um segmento é pouco viável, podemos fazer dele uma sub-rede.

6.3 A Máscara de Sub-rede

A máscara de sub-rede é um endereço de 32 bits usado para bloquear (mascarar) uma parte do endereço IP para se distinguir a parte de identificador de rede e a parte de identificador de computador (Host).

Cada computador numa rede TCP/IP precisa ter uma mascara de sub-rede (é obrigatório). Isto pode ser conseguido a partir de uma máscara padrão de classe A, B ou C (usada quando a rede não necessita de ser dividida em sub-redes) ou através de uma máscara personalizada (usada quando a rede precisa de ser dividida em sub-redes). Na

mascara padrão todos os bits que correspondem à parte do identificador da rede são colocados "1", que convertido para decimal obtêm-se o valor 255 (11111111 = 255).



Todos os bits que correspondem à parte do Host são colocados a "0", que convertido para decimal obtêm-se o valor 0 (00000000 = 0).

255	255	255	0
11111111	11111111	11111111	00000000
Rede	Rede	Rede	Host

As classes dão certa flexibilidade na distribuição dos endereços. Uma vez que o endereço IP tem tamanho fixo, uma das opções dos projetistas seria dividir o endereço IP em duas metades, dois bytes para identificar a rede e dois bytes para a estação. Entretanto isto traria inflexibilidade, pois só poderiam ser endereçados 65.536 redes, cada uma com 65.536 estações. Uma rede que possuísse apenas 100 estações estaria utilizando um endereçamento de rede com capacidade de 65536 estações, o que também seria um desperdício. *É importante conceber que dentro de uma faixa de IP há dois endereços especiais denominados endereço de rede e endereço de broadcast. O primeiro com todos os bits iguais a zero e o segundo com todos os bits iguais a um.*
Para estabelecer uma comunicação direta, servidores têm que ter os mesmos endereços de rede. Se os servidores tiverem endereços de rede diferentes, então há necessidade de usar um roteador para conectar dois segmentos de rede. *Um roteador pode conectar seguimentos de rede somente se eles tiverem endereços de rede diferentes.*

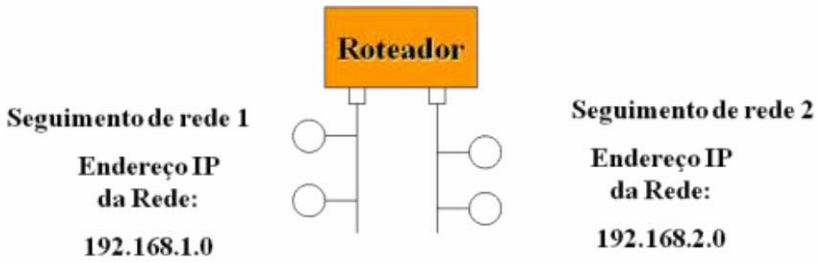


Figura 11 – Segmentos de rede

A divisão de endereçamento tradicional da Internet em classes causou sérios problemas de eficiência na distribuição de endereços. Cada rede na Internet, tenha ela 5, 200, 2000 ou 30.000 máquinas deveria ser compatível com uma das classes de endereços. Desta forma, uma rede com 10 estações receberia um endereço do tipo classe C, com capacidade de endereçar 254 estações. Isto significa um desperdício de 244 endereços. Da mesma forma, uma rede com 2.000 estações receberia uma rede do tipo classe B, e desta forma causaria um desperdício de 63.534 endereços.

O número de redes interligando-se à Internet a partir de 1988 aumentou, causando o agravamento do problema de disponibilidade de endereços na Internet, especialmente o desperdício de endereços em classes C e B. Desta forma, buscou-se alternativas para aumentar o número de endereços de rede disponíveis sem afetar o

funcionamento dos sistemas existentes. A melhor alternativa encontrada foi flexibilizar o conceito de classes - onde a divisão entre rede e host ocorre somente a cada 8 bits.

A solução encontrada foi utilizar a divisão da identificação de rede e host no endereçamento IP de forma variável, podendo utilizar qualquer quantidade de bits e não mais múltiplos de 8 bits conforme ocorria anteriormente. Um identificador adicional, a máscara, identifica em um endereço IP - qual porção dos bits é utilizada para identificar a rede e qual porção dos bits para host.

Exemplos de endereçamento

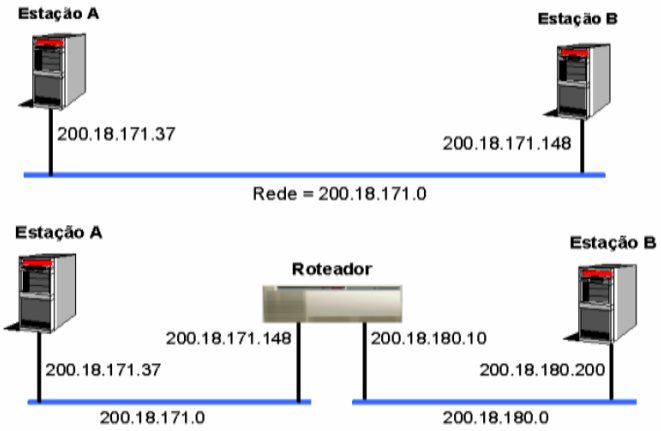


Figura 12 - Exemplos de endereçamento

A máscara pode ser compreendida também como um número inteiro que diz a quantidade de bits 1(um) utilizados. Por exemplo, uma máscara com valor 255.255.255.192, poderia ser representada como /26. Este tipo de notação é empregada em protocolos de roteamento mais recentes.

6.3.1 CONSTRUINDO SUB-REDES: Antes de tudo, é importante memorizar a tabela abaixo:

128	128	10000000
192	128 + 64	11000000
224	128 + 64 + 32	11100000
240	128 + 64 + 32 + 16	11110000
248	128 + 64 + 32 + 16 + 8	11111000
252	128 + 64 + 32 + 16 + 8 + 4	11111100
254	128 + 64 + 32 + 16 + 8 + 4 + 2	11111110
255	128 + 64 + 32 + 16 + 8 + 4 + 2 + 1	11111111

Quando se depara com uma máscara de rede e se precisa determinar o número de sub-redes, hosts válidos e endereços de broadcast que a máscara define tudo o que você tem a fazer é responder a cinco perguntas:

1. Quantas sub-redes tal máscara produz?
2. Quantos endereços de hosts válidos são obtidos por sub-rede?
3. Quais são as sub-redes válidas?
4. Quais os hosts válidos em cada sub-rede?
5. Qual o endereço de broadcast de cada sub-rede?

1) Quantas sub-redes? 2x = quantidade de sub-redes, onde "x" representa o número de bits "mascarados" ou número de "1s". Por exemplo: 11000000 (corresponde a 192 na base 10) seriam 22 = 4. Nesse caso, haveria quatro sub-redes possíveis com tal máscara.

2) Quantos hosts válidos por sub-rede? 2y – 2 = quantidade de hosts válidos, onde "y" representa o número de bits disponíveis para manipulação dos

endereços de host, ou o número de “0s”. Por exemplo: 11000000 seria 26 -2 = 62. Neste caso, existem 62 endereços válidos para hosts por sub-rede.

3) Quais são as sub-redes válidas? 256 – máscara de rede = valor da sub-rede base. A esse resultado, soma-se o valor obtido até que se atinja o número da máscara (que seria inválido). Seguindo nosso exemplo: 256 – 192 = 64 (número base e primeira sub-rede válida). 64 + 64 = 128 (segunda sub-rede válida). 128 + 64 = 192 (valor da máscara = sub-rede inválida). Portanto, as sub-redes válidas seriam 64 e 128.

4) Qual o endereço de broadcast para cada sub-rede? O endereço de broadcast seria o valor imediatamente anterior ao valor da próxima sub-rede (ou da máscara, se estivéssemos falando da última sub-rede na seqüência).

Em nosso exemplo, temos as sub-redes 64 e 128. O endereço de broadcast da primeira seria 128 – 1 =127. Já o da segunda 192 (valor da máscara) – 1 = 191.

5) Quais os hosts válidos? Os valores válidos seriam os compreendidos entre as sub-redes, menos todos os bits ligados e desligados. A melhor maneira de se identificar esses valores é se descobrindo as sub-redes válidas e os endereços de broadcast de cada uma. Em nosso exemplo, os hosts válidos estariam compreendidos nos intervalos entre 65-126 para a primeira sub-rede e 129-190 para a segunda (pois 64 e 127 são os valores que definem as respectivas.

Emprestando bits: Vamos imaginar que eu precise de uma rede para pelo menos 1.000 hosts. Como tem que ser múltiplo binário, ou seja X2, o mais próximo que temos a isso é 1024 (210). Logo, além do último octeto (8 bits) precisamos "pegar emprestado" mais 2 bits.

Máscara classe C padrão 255.255.255.0 ou 11111111.11111111.11111111.00000000;

Pegando 2 bits emprestados do terceiro octeto ficaria:

11111111 11111111 11111100 00000000 ou 255.255.252.0

Sendo os 2 primeiros octetos "fixos" da rede teríamos para sub-rede 64 para rede (26) e 1024 para host (210) o que nos daria 1024 endereços de host disponíveis para cada uma da 64 sub-redes como 1 endereço é de rede e 1 de broadcast teríamos que diminuir 2 que daria 1022 endereços validos por sub-rede.

Exemplos: Quando se deseja quer obter 2 redes lógicas de um endereço classe C , é preciso modificar a máscara de sub-rede para 255.255.255.128. Observe a figura e aplique as 5 perguntas conforme ensinado acima.

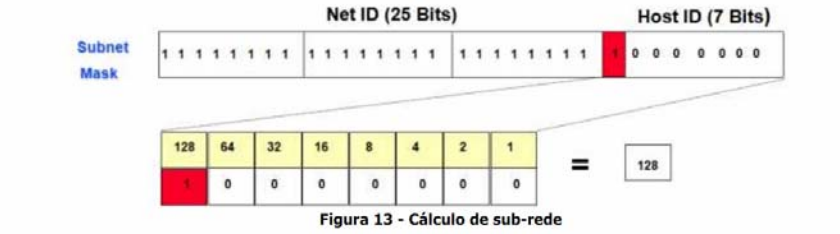


Figura 13 - Cálculo de sub-rede

Faixa total: 192.168.1.0 a 192.168.1.127
Endereço de rede: 192.168.1.0
End. Broadcast: 192.168.1.127
End. Dos hosts: 192.168.1.1 a 192.168.1.126
} Rede 0

Faixa total: 192.168.1.128 a 192.168.1.255
Endereço de rede: 192.168.1.128
End. Broadcast: 192.168.1.255
End. Dos hosts: 192.168.1.129 a 192.168.1.256
} Rede 1

Quando se deseja quer obter 4 redes lógicas de um endereço classe C , é preciso modificar a máscara de sub-rede para 255.255.255.192. Observe a figura e aplique as 5 perguntas conforme ensinado acima.

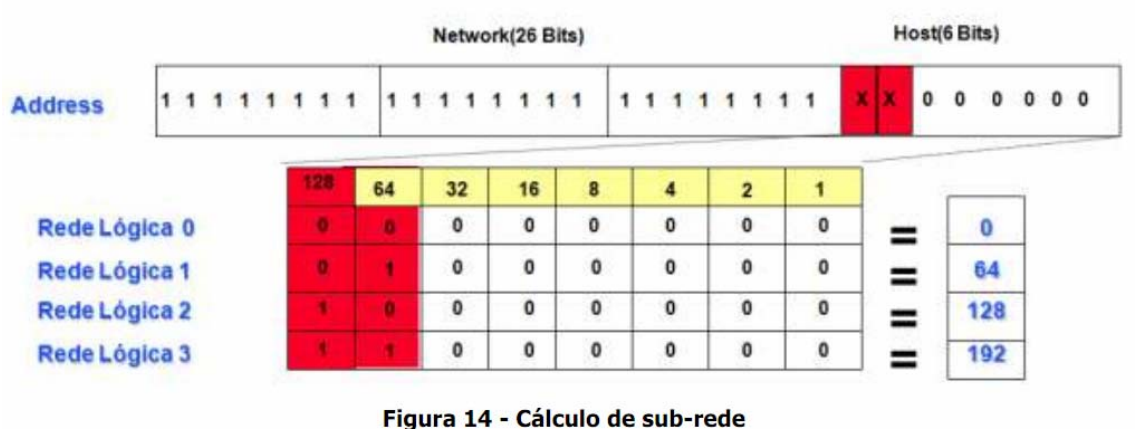


Figura 14 - Cálculo de sub-rede

Note que, para que exista comunicação entre as sub-redes é necessário um roteador, uma vez que as sub-redes não se comunicam.

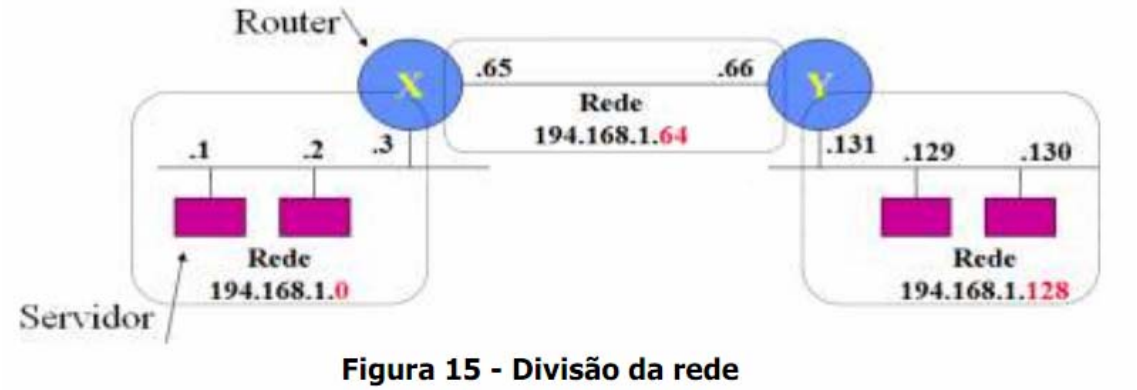


Figura 15 - Divisão da rede

Como calculado é possível usar a máscara de sub-rede 255.255.255.192 e obter 4 redes com os seguintes endereços IP:

- 192.168.1.0
- 192.168.1.64
- 192.168.1.128
- 192.168.1.192

6.3.2 CALCULANDO A REDE QUE DETERMINADO IP PERTENCE:

Este cálculo é feito pelo roteador para determinar a rede ou sub-rede para a qual um pacote deve ser enviado. Para determinar qual rede ou sub-rede host pertence é necessário realizar uma operação AND.

Exemplos:

Endereço Completo	192.168.5.10	11000000.10101000.00000101.00001010	AND
Máscara da Sub-Rede	255.255.255.0	11111111.11111111.11111111.00000000	
Rede/Sub-Rede	192.168.5.0	11000000.10101000.00000101.00000000	

Endereço Completo	192.168.5.130	11000000.10101000.00000101.10000010	AND
Máscara da Sub-Rede	255.255.255.192	11111111.11111111.11111111.11000000	
Rede/Sub-Rede	192.168.5.128	11000000.10101000.00000101.10000000	

Parte 7: A Camada de Transporte

1. A Camada de Transporte

Responsável pela movimentação de dados, de forma eficiente e confiável, entre processos em execução nos equipamentos conectados a uma rede de computadores, independentemente da rede, ou redes, física. Deve poder regular o fluxo de dados e garantir confiabilidade, assegurando que os dados cheguem a seu destino sem erros e em seqüência. Importante na camada de transporte aprender sobre os protocolos de transporte na Internet:

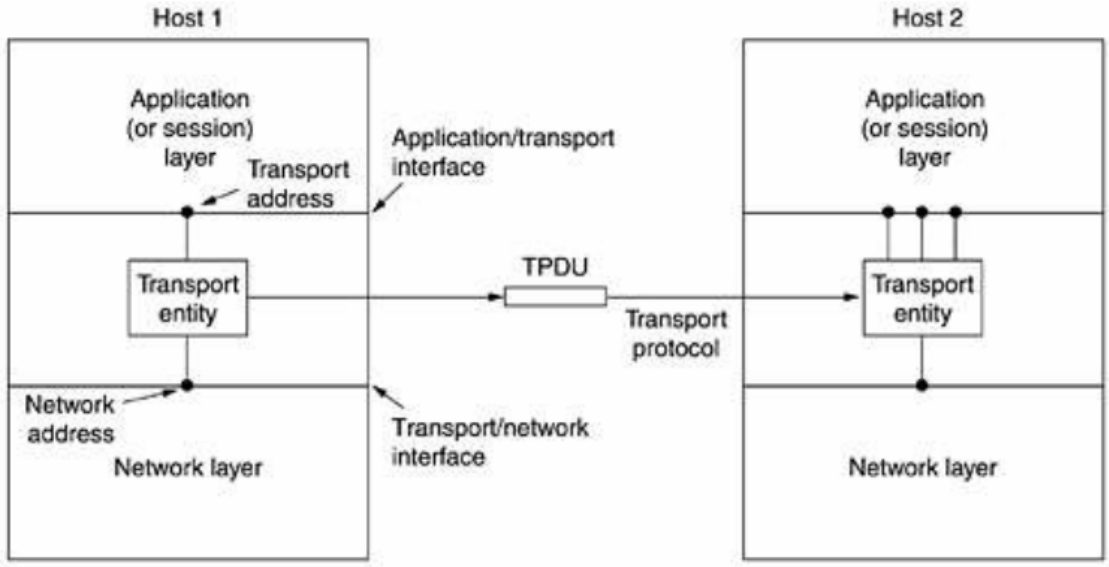
- UDP: transporte não orientado à conexão
- TCP: transporte orientado à conexão e faz controle de congestionamento do TCP

As Funções da camada de transporte:

- A Camada de transporte deve tornar transparente para os usuários, variações da
- Confiabilidade do serviço de rede;
- Transporte de unidades de dados;
- Segmentação e blocagem;
- Detecção e correção de erros fim a fim;
- Seqüenciarão;
- Controle de fluxo de dados nas conexões de transporte;
- Multiplexação (combinar várias conexões de transporte em uma mesma rede para reduzir custos) ou *splitting*;
- Transporte de dados Expressos (para sinalização).

2. Recursos da Camada de Transporte

2.1 Serviços oferecidos às camadas superiores: Seu objetivo é fornecer um serviço a camada superior confiável, eficiente e econômico, em geral, são processos presentes na camada de aplicação. Para isso, a camada de transporte utiliza vários serviços oferecidos pela camada de rede. Existe um relacionamento lógico entre as camadas de rede, de transporte e de aplicação, como ilustrado abaixo.



Assim como existem dois tipos de serviços de rede (com e sem conexão), também existe dois tipos de serviço de transporte, ambos semelhantes ao serviço de rede orientado a conexões (as conexões têm três fases: o estabelecimento, a transferência de dados e o encerramento) e sem conexão.

"A grande diferença entre os serviços na camada de rede e na camada de transporte, embora bastante semelhantes é que o código de transporte funciona inteiramente nas máquinas dos usuários, e a camada de rede funciona principalmente nos roteadores."

2.2 Qualidade de Serviço (QoS): Durante o processo de estabelecimento de uma conexão devem ser definidos alguns detalhes sobre como a mesma transcorrerá. Um desses detalhes é a definição da qualidade de serviço (QoS), a qual, na prática, define os limites mínimos aceitáveis para certos parâmetros de desempenho para que a conexão seja mantida. Os valores estabelecidos de QoS durante a fase de conexão são, portanto, pisos de desempenho para a conexão. Dentro do protocolo OSI se definem diversas classes de QoS, segundo patamares de erros e perdas de pacotes. A classe de uma rede (do ponto de vista de qualidade) é definida pela classificação ISO. Além dos tipos de classe do protocolo OSI ainda se definem, durante o estabelecimento da conexão, parâmetros de qualidade como:

- Atraso de estabelecimento de conexão;
- Probabilidade de falha de estabelecimento de conexão;
- Vazão (velocidade) da rede;
- Atraso de trânsito;
- Prioridade;
- Resiliência;
- Taxa de erros residuais;

2.3 Endereçamento: O problema de endereçamento consiste em permitir que várias conexões sejam estabelecidas, por uma mesma máquina, com uma ou mais máquinas simultaneamente. Isso implica em que cada conexão deverá ter endereços específicos, que permitam à camada de transporte diferenciar entre uma conexão e outra. Isso é feito através de endereços de portas de serviço, conhecidas por TSAP (*Transport Service Access Point*). As entidades que querem trocar informações devem, então, especificar endereços TSAP em que efetivarão tais trocas. Em situações específicas, como acesso a um sistema de arquivos, esse processo de endereçamento não funciona corretamente. Nesses casos, o que se faz é obter o endereço TSAP do sistema de arquivos através de uma conexão estabelecida entre a entidade que quer se conectar a ele e a um servidor de nomes, que realiza suas conexões em uma TSAP previamente conhecida, como ocorre, por semelhança, ao serviço de telnet, que funciona na porta 22.

3. Conexões

3.1 Estabelecimento de uma conexão: Estabelecer uma conexão pode ser algo bastante complicado, uma vez que neste processo podem ocorrer perdas ou duplicações de pacotes. Em uma sub-rede congestionada, onde as confirmações quase não chegam a tempo, o pacote pode ser transmitido várias vezes. Onde se utiliza datagrama, podem ocorrer retardos em alguns nós e só surgirem no destino minutos depois.

A pior situação é a criação de um circuito virtual, a transmissão de informações ocorre, mas alguns pacotes ficaram retidos em algum roteador, houve sua duplicação e após encerrada a conexão esses pacotes continuam chegando.

Para solucionar esses problemas, uma solução é utilizar endereços de transporte descartáveis ou atribuir a cada conexão um identificador de conexões, não sendo tão eficiente uma vez que há a necessidade da entidade de transporte manter um histórico das conexões por tempo indeterminado.

Para cuidar dessa questão foi criado o protocolo de *handshake* de três vias no estabelecimento da conexão. Isso deve ocorrer antes que quaisquer dados sejam transferidos entre os dois *hosts*. A figura representa o cliente ou *host* de origem iniciando uma conexão com o servidor ou host de destino. O termo cliente é usado para significar o host que solicita algum tipo de serviço de outro host. Um servidor é um host que atende, em um número de porta conhecido, requisições de um serviço específico. O TCP exige que uma porta de destino ou serviço sejam especificados. Exemplos de portas de destino podem ser 23 (*Telnet*), 25 (correio eletrônico) ou 80 (também conhecida como a porta de servidor HTTP ou *Web*). O *handshake* de três vias acontece da seguinte maneira:

- O cliente envia um flag SYN para sinalizar uma requisição de uma conexão TCP com o servidor.
- Se o servidor estiver em execução e oferecer o serviço desejado (e se puder aceitar a conexão chegando), ele envia ao cliente sua própria requisição de conexão sinalizada por um novo SYN e acusa o recebimento da requisição de conexão do cliente com um flag ACK. Tudo isso é feito em um único pacote.
- Finalmente, se o cliente receber os flags SYN e ACK, e ainda quiser continuar a conexão, ele envia um único ACK final ao servidor. Isso avisa que o cliente recebeu a requisição de conexão do servidor.

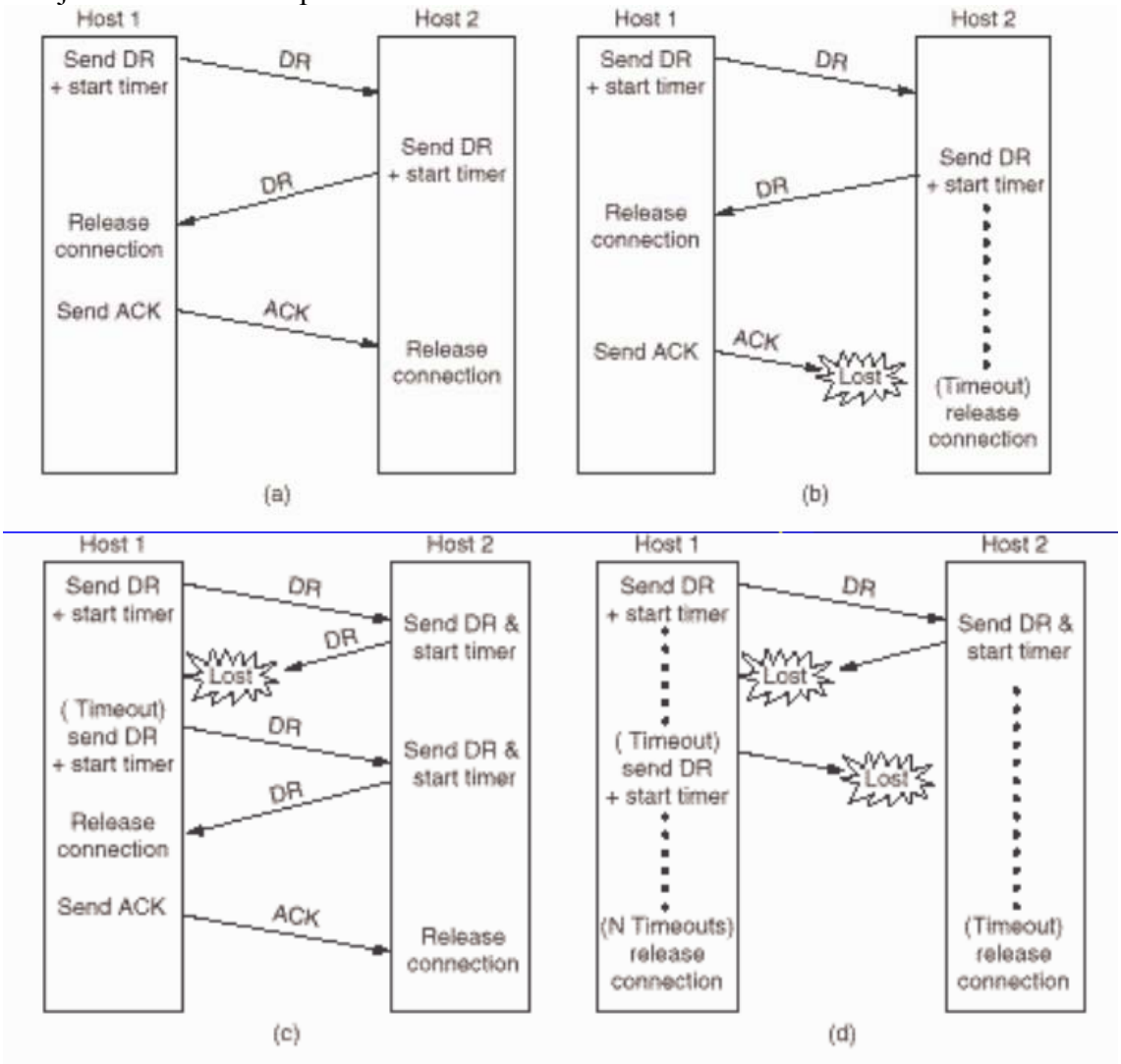
Após o **handshake de três vias** (Tomlison, 1975) ter sido executado dessa maneira, a conexão está estabelecida. Agora os dados podem ser trocados entre os dois hosts.

3.2 Encerramento de uma conexão: Encerrar uma conexão é um tanto mais fácil que estabelecê-la. Isto pode acontecer de duas maneiras: do modo elegante ou do modo grosseiro. O método elegante seria o mesmo que dizer em uma conversa telefônica: "Obrigado, mas não estamos interessados", e desligar o telefone. Isso informa à operadora de telemarketing que a conversa acabou, e que ela agora deve desligar e efetuar outra ligação inconveniente na hora do jantar. O método grosseiro seria simplesmente desligar o telefone.

Quando o término elegante da sessão TCP é utilizado, um dos hosts, o cliente ou o servidor, sinaliza, com um flag FIN para o outro, que ele deseja terminar a sessão. O host receptor sinaliza de volta com um flag ACK (ou seja, acusa o recebimento da

requisição). Isso finaliza apenas metade da conexão. Então, o outro host também precisa iniciar um FIN, e o host receptor precisa acusar seu recebimento.

O método grosseiro, o segundo método de término é um encerramento abrupto da conexão. Isso é feito com um host enviando ao outro um flag RESET, que sinaliza o desejo de terminar abruptamente a conexão.



3.3 Multiplexação: A multiplexação de várias conversações em conexões, circuitos virtuais e enlaces físicos tem um papel importante em diversas camadas da arquitetura de rede. Ela pode surgir de diversas formas, como por exemplo, se houver apenas um endereço de rede disponível em um host, todas as conexões nessa máquina terão que utilizá-la.

São classificadas de duas formas:

- **Ascendente:** Várias conexões de transporte na mesma conexão de rede. Motivo: Tarifação.
- **Descendente**: Várias conexões de rede e distribuição do tráfego por uma conexão de transporte. Motivo: Necessidade de grande largura de banda.

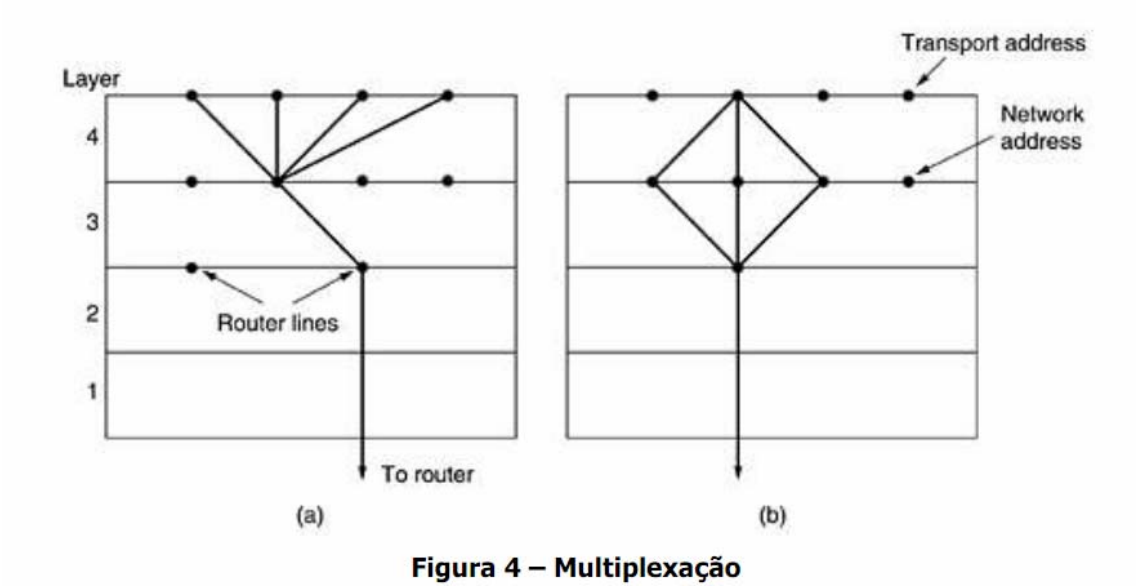


Figura 4 – Multiplexação

3.4 Recuperação de Falhas: Na recuperação de falhas a primeira consideração a ser feita é sobre qual o tipo de sub-rede: se datagrama ou circuito-virtual. Se a camada de rede provê um serviço de datagramas as entidades de transporte esperam a todo o momento os TPDU's perdidos, e mesmo por rotas distintas eles sempre chegarão.

Se a conexão for orientada a serviço a perda do circuito será manipulada estabelecendo uma nova conexão para receber os TPDU's perdidos.

Problemas como este mostra à questão de uma implementação de confirmação fim a fim. Em princípio, o protocolo de transporte é fim a fim, pois não é encadeado como as camadas inferiores. Imaginemos um usuário que solicita transações relativas a um banco de dados remoto. Vamos supor que a entidade de transporte remota estiver programada de modo a passar primeiro as TPDUs para a camada imediatamente superior e só então enviar a confirmação.

Mesmo nesse caso, o fato de uma confirmação ter sido recebida na máquina do usuário não quer dizer necessariamente que o host remoto funcionou por tempo suficiente para atualizar o banco de dados. Uma confirmação fim-a-fim verdadeira, cujo recebimento indica que o trabalho foi realmente realizado e cuja falta indica que ele não foi cumprido, talvez seja *algo impossível de se alcançar.*

4. UDP (User Datagram Protocol)

Muito usado por aplicações de multimídia contínua (*streaming*):

- Tolerantes à perda;
- Sensíveis à taxa;

O conjunto de protocolos da Internet admite um protocolo de transporte sem conexões, UDP (*User Datagram Protocol*). Este oferece um meio para as aplicações enviarem datagramas IP encapsulados sem que seja necessário estabelecer uma conexão.

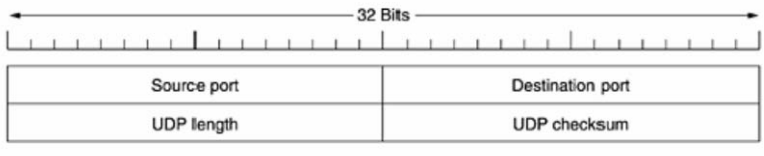


Figura 6 - Segmento UDP

O UDP transmite segmentos que consistem em um cabeçalho de 8 bytes, seguido pela carga útil, ou seja, o UDP é basicamente o IP com um pequeno cabeçalho.

- As duas portas indicadas na figura acima servem para identificar os pontos extremos (máquinas de origem e destino).
- A porta de origem é usada principalmente quando uma resposta dever ser devolvida à origem.
- O campo UDP *length* inclui o cabeçalho de 8 *bytes* e os dados.
- O campo UDP *checksum* é opcional e armazenado como 0 se não for calculado (um valor 0 verdadeiro calculado é armazenado com todos os bits iguais a 1). É importante saber que UDP não realiza. Ele não realiza controle de fluxo, controle de erros ou retransmissão após a recepção de um segmento incorreto. Tudo isso cabe aos processos do usuário. O que ele faz é fornecer uma interface para o protocolo IP com o recurso adicional de demultiplexação de vários processos que utilizam as portas.

Uma área na qual o UDP é especialmente útil é a de situações cliente/servidor. Com freqüência, o cliente envia uma pequena solicitação ao servidor e espera uma pequena resposta de volta. Outra aplicação muito comum para o UDP é o DNS (*Domain Name System*), um programa que precisa pesquisar o endereço IP de algum nome de host - por exemplo, www.dfiori.com.br - pode enviar um pacote UDP contendo o nome do host a um servidor DNS. O servidor responde com um pacote UDP que contém o endereço IP.

4.1 Demultiplexação não orientada a conexão:

Cria *sockets* com números de porta, exemplo:

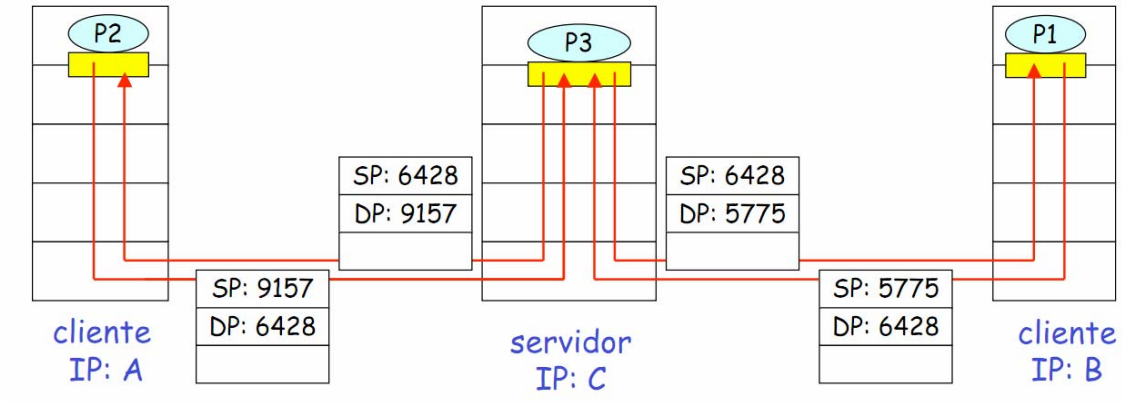
```
DatagramSocket mySocket1 = new DatagramSocket(99111);
DatagramSocket mySocket2 = new DatagramSocket(99222);
```

Socket UDP identificado por dois valores: (endereço IP de destino, número da porta de destino).

Quando o hospedeiro recebe o segmento UDP:

- Verifica o número da porta de destino no segmento.
- Direciona o segmento UDP para o socket com este número de porta Datagramas com IP de origem diferentes e/ou portas de origem diferentes são direcionados para o mesmo *socket*.

```
DatagramSocket serverSocket = new DatagramSocket(6428);
```



5. TCP (Transmission Control Protocol)

5.1 Modelo de Serviço TCP: O serviço TCP é obtido quando tanto o transmissor quanto o receptor criam pontos extremos chamados soquetes. Cada soquete tem um número de soquete (endereço) que consiste no endereço IP do host e em um número de 16 *bits* local para esse *host*, chamado porta. Para que o serviço TCP funcione, é necessário que uma conexão seja explicitamente estabelecida entre um soquete da máquina transmissora e um soquete da máquina receptora.

Cabe ressaltar que um soquete pode ser utilizado por várias conexões ao mesmo tempo. Em outras palavras, duas ou mais conexões podem terminar no mesmo soquete.

As portas com números abaixo de 1024 são denominadas portas conhecidas e são reservadas para serviços padrão. Por exemplo, qualquer processo que deseje estabelecer uma conexão com um *host* para transferir um arquivo usando FTP pode se conectar à porta 21 do *host* de destino e entrar em contato com seu *daemon* de FTP. A lista de portas conhecidas é dada em www.iana.org.

Porta	Protocolo	Uso
21	FTP	Transferência de arquivos
23	Telnet	Login remoto
25	SMTP	Correio eletrônico
69	TFTP	Protocolo trivial de transferência de arquivos
79	Finger	Pesquisa de informações sobre um usuário
80	HTTP	World Wide Web
110	POP-3	Acesso remoto a correio eletrônico
119	NNTP	Notícias da USENET

Figura 5 - Portas e Protocolos

5.2 Características do Serviço TCP:

- Todas as conexões TCP são Full-duplex e ponto a ponto
- O TCP não admite os processos de multidifusão e difusão.
- Uma conexão TCP é um fluxo de bytes e não um fluxo de mensagens.

Quando uma aplicação repassa dados para a entidade TCP, ela pode enviá-las imediatamente ou armazená-las em um buffer (para aguardar outros dados e enviar um volume maior de uma só vez). Entretanto, há ocasiões em que a aplicação realmente quer que os dados sejam enviados de imediato. Por exemplo, um usuário que tenha se conectado a uma máquina remota. Depois que uma linha de comandos é preenchida e a tecla Enter (ou *Carriage Return*) é pressionada, é essencial que a linha seja transportada à máquina remota imediatamente.

Outro recurso do serviço TCP é o dos dados urgentes. Quando um usuário interativo pressiona a tecla DEL ou as teclas CTRL- C para interromper um processo remoto já iniciado, a aplicação transmissora adiciona algumas informações de controle ao fluxo de dados e o entrega ao TCP juntamente com um flag URGENT. Isso faz com que o serviço TCP pare de acumular dados e transmita tudo imediatamente.

Uma característica fundamental do TCP que domina o projeto do protocolo é que cada byte em uma conexão TCP tem seu próprio número de seqüência de 32 bits. As entidades transmissoras e receptoras do TCP trocam dados na forma de segmentos.

O protocolo básico utilizado pelas entidades TCP é o protocolo de janela deslizante. Quando é enviado um segmento, o transmissor também dispara um *timer*. Quando o segmento chega ao destino, a entidade TCP receptora retorna um segmento (com

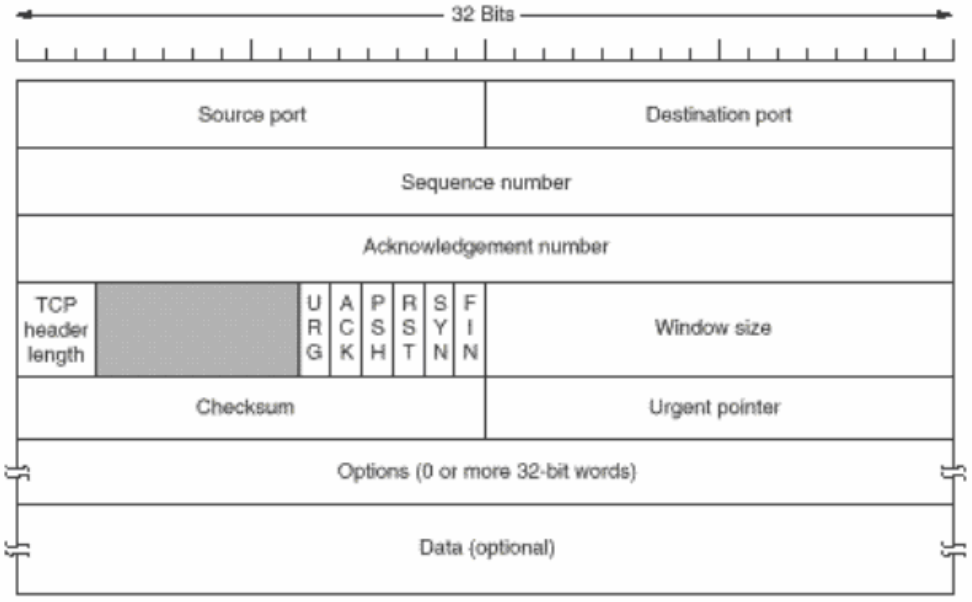
ou sem dados, de acordo com as circunstâncias) com um número de confirmação igual ao próximo número de seqüência que espera receber. Se o timer do transmissor expirar antes da confirmação ser recebida, o segmento será retransmitido.

5.3 Primitivas do Socket TCP:

Primitive	Meaning
SOCKET	Create a new communication end point
BIND	Attach a local address to a socket
LISTEN	Announce willingness to accept connections; give queue size
ACCEPT	Block the caller until a connection attempt arrives
CONNECT	Actively attempt to establish a connection
SEND	Send some data over the connection
RECEIVE	Receive some data from the connection
CLOSE	Release the connection

Fig. 6-6. The socket primitives for TCP.

5.4 O Quadro TCP:



5.5 Demultiplexação (DEMUX) orientado a conexão:

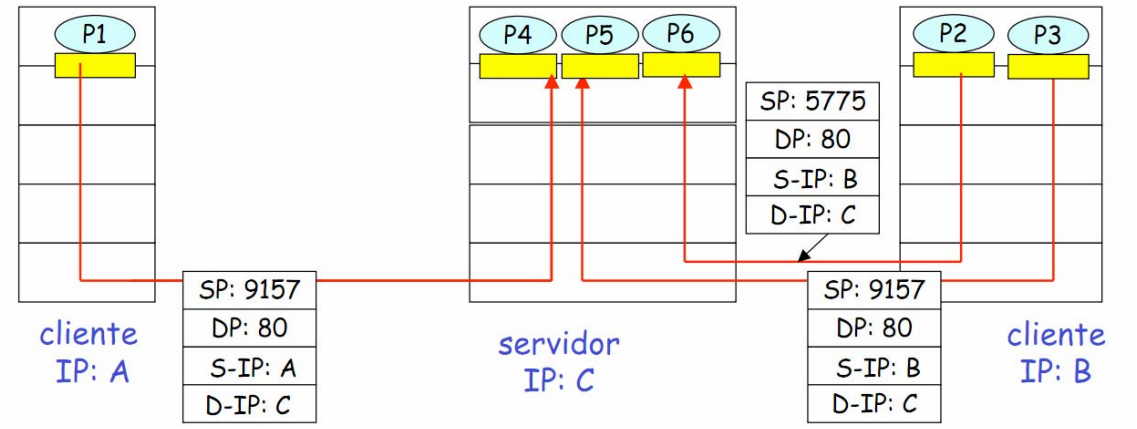
Socket TCP identificado por 4 valores:

- Endereço IP de origem
- End. porta de origem
- Endereço IP de destino
- End. porta de destino

O Hospedeiro receptor usa os quatro valores para direcionar o segmento ao *socket* apropriado.

O Hospedeiro servidor pode suportar vários *sockets* TCP simultâneos:

- Cada socket é identificado pelos seus próprios 4 valores
- Servidores Web possuem sockets diferentes para cada cliente conectado
- HTTP não persistente terá um socket diferente para cada requisição.



6. Questões Relativas a desempenho de UDP e TCP em redes sem Fio

Na teoria, os protocolos de transporte deveriam ser independentes da tecnologia da camada de rede em que se baseiam. O TCP não deveria se preocupar com o fato do IP estar sendo executado sobre fibra ou rádio. Na prática, isso é importante, pois a maioria das implementações do TCP foi otimizada com todo o cuidado, de acordo com suposições que são verdadeiras para as redes fisicamente conectadas, mas que falham no caso das redes sem fios. Ignorar as propriedades da transmissão sem fio pode levar a uma implementação do TCP logicamente correta, mas que tem um desempenho catastrófico.

O principal problema é o algoritmo de controle de congestionamento. Quase todas as implementações do TCP atuais pressupõem que os *timeouts* ocorrem devido a congestionamentos e não a pacotes perdidos. Conseqüentemente, quando um *timer* expira, o TCP diminui o ritmo e começa a transmitir de modo mais lento. A idéia por trás dessa abordagem é reduzir a carga da rede e assim diminuir o congestionamento. No entanto, os enlaces de dados das transmissões sem fios não são confiáveis. Apesar de o UDP não ter os mesmos problemas do TCP, a comunicação sem fio também cria algumas dificuldades para ele.

O principal problema é que os programas utilizam o UDP esperando que ele seja altamente confiável. Para programas que conseguem se recuperar de mensagens UDP perdidas há apenas um custo considerável. Porém, a passagem repentina de um ambiente no qual as mensagens teoricamente podem se perder, mas raras vezes se perdem, para um ambiente em que a perda de mensagens é constante pode ter um efeito catastrófico sobre o desempenho.

Parte 8: A Camada de Aplicação

1. A Camada de Aplicação

Depois de estudar todas as camadas preliminares, chegamos à camada onde são encontradas todas as aplicações. As camadas situadas abaixo da camada de aplicação têm a função de oferecer um serviço de transporte confiável, mas, na verdade, elas não executam qualquer tarefa para os usuários, dependem de algum software capaz de dar suporte a algum tipo de serviço.

2. DNS – Domain Name System

Sabe-se que quando se deseja algum recurso, este precisa ser referenciado através de seu endereço binário, ou seja, seu endereço IP, que são dif íceis de memorizar. Exemplo:

> Enviar um e-mail usando o domínio do e-mail um número binário, como aluno@201.10.20.31. Conseqüentemente, foram introduzidos nomes em ASCII para desacoplar os nomes das máquinas dos endereços desses endereços numéricos. Para isso, é necessário um mecanismo que faça este mapeamento.

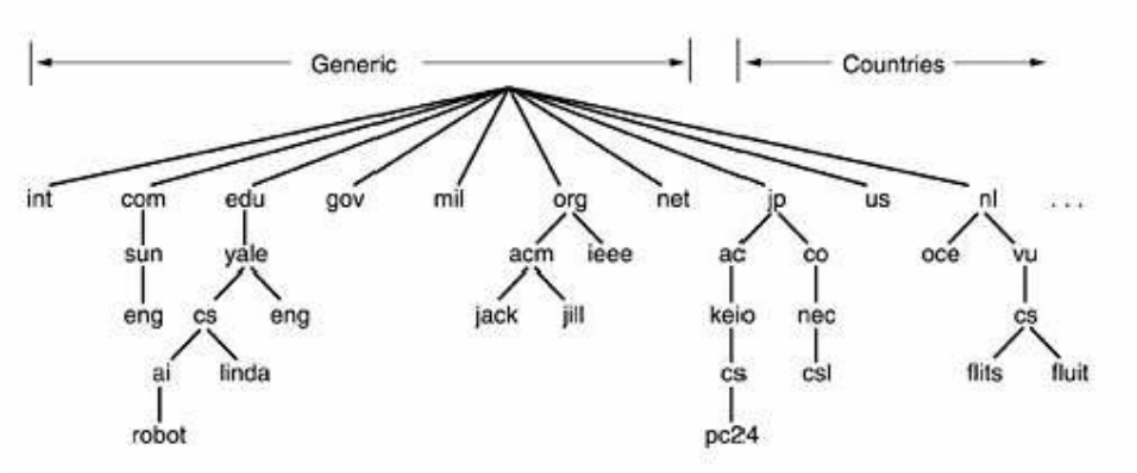
Na ARPANET o mapeamento era realizado através de um arquivo chamado hosts.txt, que listava todos os hosts e seus endereços IP, no entanto para uma rede demasiadamente grande esta abordagem é inviável.

Diante dessa mudança de perspectiva foi criado o DNS (*Domain Name System* – Sistema de Nome de Domínios). A essência do DNS é a criação de um esquema hierárquico de atribuição de nomes baseado no domínio e em um sistema de bancos de dados distribuídos para implementar esse esquema de nomenclatura. Ele é usado principalmente para mapear nomes de hosts e destinos de mensagens de correio eletrônico em endereços IP, mas também pode ser usado para outros objetivos. O DNS é definido nas RFCs 1034 e 1035.

Para mapear um nome em um endereço IP, um programa aplicativo chama um procedimento de biblioteca denominado resolvedor e repassa a ele o nome como um parâmetro. O resolvedor envia um pacote UDP a um servidor DNS local, que procura o nome e retorna o endereço IP ao resolvedor. Em seguida, o resolvedor retorna o endereço IP ao programa aplicativo que fez a chamada. Munido do endereço IP, o programa pode então estabelecer uma conexão TCP com o destino ou enviar pacotes UDP. Uma analogia que pode ser feita é com relação ao sistema postal.

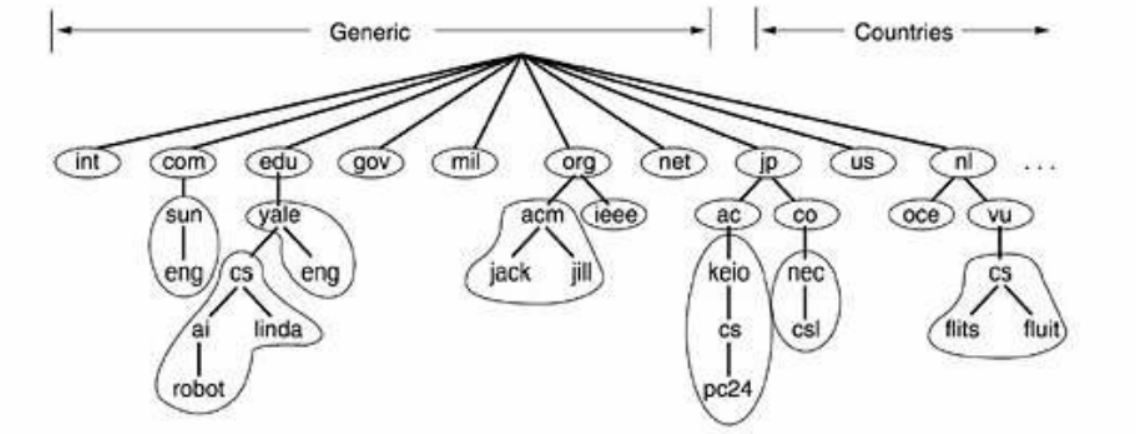
Conceitualmente, a Internet é dividida em mais de 200 domínios de nível superior, onde cada domínio cobre muitos hosts. Cada domínio é particionado em subdomínios, que também são particionados e assim por diante. Todos esses domínios podem ser representados por uma árvore, como na figura abaixo. As folhas da árvore representam domínios que não têm subdomínios (mas que contêm máquinas, é claro).

Um domínio folha contém um único host ou pode representar uma empresa e conter milhares de hosts. Existem dois tipos de domínios de nível superior: genéricos e de países. Os domínios genéricos originais eram com (comercial), edu (instituições educacionais), gov (inst ituições governamentais), int (certas organizações internacionais), mil (órgãos das forças armadas), net (provedores de rede) e org (organizações sem fins lucrativos). Os domínios de países incluem uma entrada para cada país, conforme a definição da ISO 3166.



Em geral, é fácil obter um domínio de segundo nível, como <nome-da-empresa.com>. Isso exige apenas um registro do domínio de nível superior correspondente (nesse caso, com) para verificar se o nome desejado está disponível e não é marca registrada de outra pessoa. Para que um novo domínio seja criado, é necessária a permissão do domínio no qual ele será incluído.

É um Servidor de Nomes pelo menos na teoria, um único servidor de nomes poderia conter o banco de dados DNS inteiro e responder a todas as consultas referentes ao banco. Na prática, esse servidor ficaria tão sobrecarregado que seria inútil. Além disso, caso esse servidor viesse a ficar fora do ar, toda a Internet seria atingida. Para evitar os problemas associados à presença de uma única fonte de informações, o espaço de nomes do DNS é dividido em zonas não superpostas.



Parte 9: A Segurança das Redes

1. Segurança das Redes

A maior parte dos problemas de segurança é causada intencionalmente por pessoas maliciosas que tentam obter algum benefício, chamar a atenção ou prejudicar alguém. Para tornar uma rede segura, com freqüência é necessário lidar com adversários inteligentes, dedicados e, às vezes, muito bem subsidiados.

Os problemas de segurança das redes podem ser divididos nas seguintes áreas interligadas:

- Confidencialidade;

- Integridade;
- Disponibilidade;
- Autenticidade;
- Não repúdio.

Muitas são as soluções apresentadas sobre a camada de aplicação, no entanto é importante considerar a que parte da pilha de protocolos pertence a segurança de redes.

Provavelmente não existe uma parte específica. Todas as camadas contribuem de alguma forma.

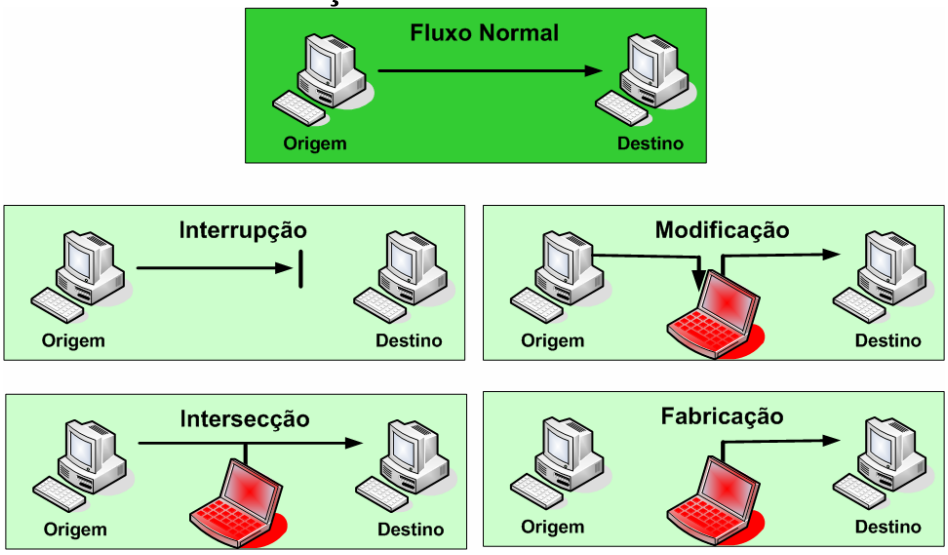
Na camada de enlace de dados, os pacotes de uma linha ponto a ponto podem ser codificados à medida que saem de uma máquina, e decodificados quando entram em outro sistema. No entanto, essa solução se mostra ineficiente quando os pacotes têm de atravessar vários roteadores, pois é necessário decriptografar os pacotes em cada roteador, o que os torna vulneráveis a ataques dentro do roteador.

Na camada de rede, podem ser instalados firewalls para manter ou descartar pacotes.

A segurança do IP também funciona nessa camada.

Na camada de transporte, é possível criptografar conexões inteiras fim a fim, ou seja, processo a processo. Para obter segurança máxima, é necessário que ela seja fim a fim. Finalmente, questões como autenticação do usuário e não repúdio só podem ser tratadas na camada de aplicação.

2. Problemas na comunicação:



2.1 Interrupção: Ataque passivo, isto é não manipula a informação tragegada. Realiza o que chamamos de DoS (Deny of service) ou negação de serviços. Se for um DDoS é o ataque distribuído de negação de serviços.

2.2 Modificação: Ataque ativo, no qual há manipulação de informação, o atacante ouve o canal e realiza as modificações das informações enviando ao destinatário as informações manipuladas. Esse ataque recebe o nome de *man in the middle (MIM).*

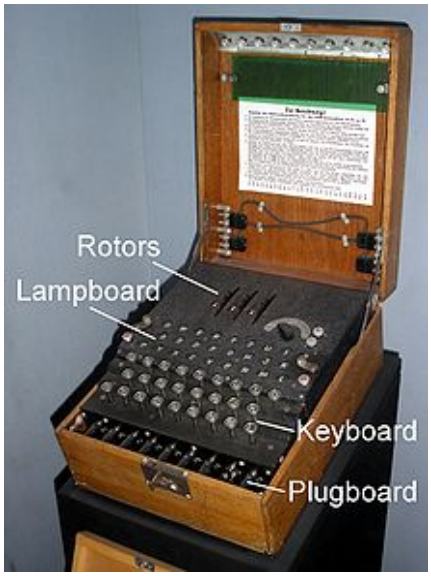
2.3 Intersecção: Ataque passivo, atacante não manipula as informações porém as escuta e verifica seu conteúdo. Esse ataque é chamado de *sniffer.*

2.4 Fabricação: Ataque Ativo, no qual há manipulação de informações, o atacante gera informações se fazendo como o remetente para o destinatário.

3. Criptografia

A palavra criptografia tem origem no grego e significa "Palavra Oculta". Júlio César escrevia textos criptografados para Cícero e para seus generais a mais de 2.000 anos atrás, usando um cifrador onde cada letra era substituída por uma deslocada três posições no alfabeto. Thomas Jefferson utilizou um equipamento chamado Roda Criptográfica para manter comunicações privadas quando foi representante junto ao governo Francês (1784-1789) porque na época, os serviços de correio abriam toda a correspondência enviada ou recebida.

No século 20 a máquina Enigma foi um dos segredos mais bem guardados na Segunda Grande Guerra, usada pelos Alemães para proteger as comunicações entre o comando e as embarcações navais. Na figura abaixo podemos visualizar um exemplar de uma máquina Enigma.



3.1 Conceitos de Criptografia

Quando alguns profissionais falam de criptografia, eles costumam fazer distinção entre cifras e códigos.

- **Cifra** é uma transformação de caractere por caractere ou de bit por bit, sem levar em conta a estrutura lingüística da mensagem.
- **Código** substitui uma palavra por outra palavra ou símbolo. Os códigos não são mais utilizados, embora tenham uma história gloriosa.
- **Criptografia**: é o processo de converter um texto aberto em um texto cifrado;
- **Criptografia Reversa ou Decriptografia**: é o processo de reconverter um texto cifrado em texto aberto;
- **Chave**: Conceito relacionado ao código (segredo) do processo criptográfico.

Tanto a criptografia, quanto a decriptografia utilizam chave(s) e um algoritmo(s).

Dois são os tipos de criptografia quanto à utilização da chave:

- Simétrica;
- Assimétrica

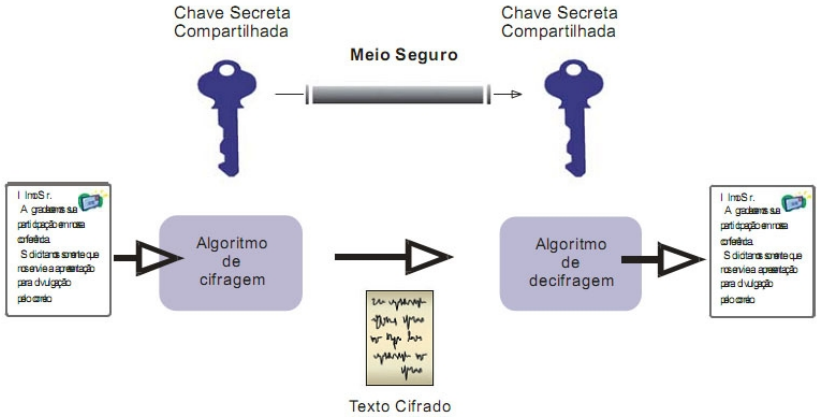
3.2 Criptografia simétrica

Este modelo de criptografia requer uma chave compartilhada. Sua performance é rápida e segura, contudo não é prático para um número grande de usuários devido a necessidade de se compartilhar a chave.

É importante notar que nas aplicações apresenta melhor resultado, porém só deve ser usada sempre que a performance for o fator determinante .

Exemplos:

DES, IDEA, Blowfish, Twofish, RC2, RC4.



3.3 Criptografia Assimétrica: Assinaturas de chave pública

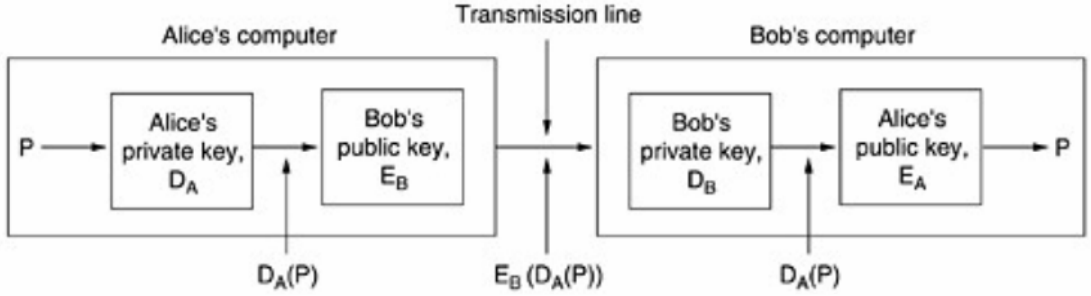
Um problema estrutural com o uso da criptografia de chave simétrica para assinaturas digitais é que todos têm de confiar. Além disso, tem de ler todas as mensagens assinadas. Os candidatos mais lógicos à execução do servidor são o governo, os bancos, os contadores e os advogados. Infelizmente, nenhuma dessas organizações inspira total confiança a todos os cidadãos. Daí, seria interessante se o ato de assinatura de documentos não exigisse a presença de uma autoridade confiável.

Felizmente, a criptografia de chave pública pode trazer uma importante contribuição para esse caso. Vamos supor que os algoritmos de criptografia e descriptografia de chave pública tenham a propriedade de que E(D(P)) = P além, é claro, da propriedade habitual de que D(E(P)) = P . (O RSA tem essa propriedade; portanto, a suposição não é irracional.) Supondo-se que seja esse o caso, Alice pode enviar uma mensagem de texto simples assinada, P, para Bob transmitindo EB(DA (P)).

Observe que Alice conhece sua própria chave de descriptografia (privada), DA, assim como a chave pública de Bob, EB ; portanto, a criação dessa mensagem é algo que Alice pode fazer.

Quando recebe a mensagem, Bob a transforma usando sua chave privada e produz DA (P), como mostra a Figura abaixo. Ele guarda esse texto em um lugar seguro e depois aplica EA para obter o texto simples original.

A figura exibe as assinaturas digitais com o uso da criptografia de chave pública



Para ver como a propriedade de assinatura funciona, suponha que posteriormente Alice negue ter enviado a mensagem P para Bob. Quando o caso chegar aos tribunais, Bob poderá produzir tanto P quanto DA (P). O juiz pode confirmar com facilidade que Bob certamente tem uma mensagem válida criptografada por DA simplesmente aplicando EA à mensagem. Como Bob não sabe qual é a chave privada de Alice, a única forma de Bob ter adquirido uma mensagem criptografada por essa chave seria se Alice de fato a tivesse enviado. Enquanto estiver presa por perjúrio e fraude, Alice terá bastante tempo para inventar novos algoritmos de chave pública muito interessantes.

Apesar da utilização da criptografia de chave pública para assinaturas digitais ser um esquema elegante, existem problemas relacionados ao ambiente no qual elas operam e não ao algoritmo básico. De um lado, Bob só poderá provar que uma mensagem foi enviada por Alice enquanto *DA* permanecer secreta. Se Alice revelar sua chave secreta, o argumento deixará de existir, pois qualquer um poderia ter enviado a mensagem, inclusive o próprio Bob.

Por exemplo, pode ocorrer um problema se Bob for o corretor de ações de Alice. Alice solicita a Bob que ele compre ações ou títulos de uma determinada empresa. Logo depois disso, o preço cai abruptamente. Para repudiar a mensagem que enviou a Bob, Alice vai à polícia afirmando que sua casa foi assaltada, e o PC que continha sua chave foi roubado. Dependendo das leis do estado ou do país onde mora, ela poderá ou não ser legalmente processada, em especial se afirmar que só descobriu o roubo quando chegou em casa após o trabalho, muitas horas depois do ocorrido.

Outro problema com o esquema de assinatura é o que acontecerá se Alice decidir alterar sua chave. Isso é legal, e provavelmente é uma boa idéia fazê-lo de vez em quando. Se mais tarde surgir um caso jurídico, como descrevemos antes, o juiz aplicará a EA atual a DA (P) e descobrirá que ela não produz P . Nesse momento, a situação de Bob ficará complicada.

4. Gerenciamento de chaves públicas

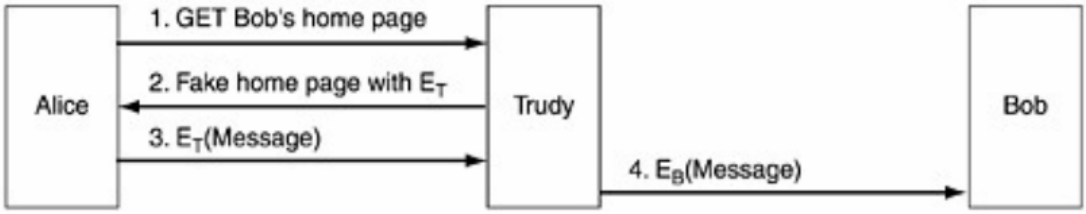
A criptografia de chave pública torna possível a comunicação segura para pessoas que não compartilham uma chave comum, e também possibilita a assinatura de mensagens sem a presença de uma terceira parte confiável. Finalmente, os sumários de mensagens assinados permitem verificar com facilidade a integridade de mensagens recebidas.

Porém, existe um problema que ignoramos até aqui: se Alice e Bob não conhecem um ao outro, como ele irão obter as respectivas chaves públicas para iniciar o processo de comunicação? A solução óbvia — colocar a chave pública no *Website* — não funciona pela seguinte razão: suponha que Alice queira pesquisar a chave pública de Bob em seu Web site. Como ela fará isso? Bem, Alice começa digitando o URL de Bob. Seu navegador então pesquisa o endereço DNS da home page de Bob e envia a ele uma solicitação GET.

Infelizmente, Trudy intercepta a solicitação e responde com uma home page falsa, talvez uma cópia da home page de Bob, exceto pela substituição da chave pública de Bob pela chave pública de Trudy.

Quando Alice codifica sua primeira mensagem com ET, Trudy a decodificará, lerá e recodificará com a chave pública de Bob, enviando a mensagem a Bob, que não sabe que Trudy está lendo suas mensagens recebidas. Pior ainda, Trudy poderia modificar as mensagens antes de recodificá-las para Bob. É claro que há necessidade de algum mecanismo para garantir que as chaves públicas possam ser trocadas em segurança.

Um modo de Trudy subverter a criptografia de chave pública



Bibliografia:
- Tanenbaum Andrew S. e Wetherall, “Redes de Computadores”, Editora PEARSON, 5° Edição, edição traduzida (2010).
- Tanenbaum Andrew S., “Redes de Computadores”, Editora Campus, 3° Edição, Versão traduzida (1997).
- Kurose James F.,Ross Keith W., “Computer Networking: A Top-Down Approach”, Addison-Wesley Publishing Company, USA, 5th edition (2009).
- Comer E. Douglas, Stevens L. David, “Interligação em rede com TCP/IP – Projeto, implementação e detalhes internos”, Volume II, Editora Campus, Versão traduzida, 3° Edição (1998).
-Aulas do Professor Ricardo Rodrigues Barcerlar http://www.ricardobarcelar.com.br