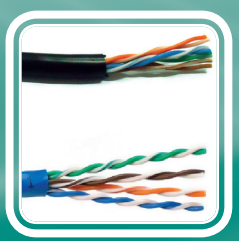
rede e-Tec Brasil

Redes de Computadores II

Marcos Prado Amaral

Curso Técnico em Planejamento e Gestão em Tecnologia da Informação

CEFET-MG

UNIVERSIDADE FEDERAL DE SANTA CATARINA

PDE | PRONATEC
PROGRAMA NACIONAL DE ACESSO AO ENSINO TÉCNICO E EMPREGO

Ministério da Educação

Redes de Computadores II

Marcos Prado Amaral

Belo Horizonte - MG

2012

Presidência da República Federativa do Brasil

Ministério da Educação

Secretaria de Educação Profissional e Tecnológica

Este Caderno foi elaborado em parceria entre o Instituto Federal de Educação, Ciência e Tecnologia do Piauí e a Universidade Federal de Santa Catarina para a Rede e-Tec Brasil.

Equipe de Elaboração
Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG

Coordenação Institucional
José Wilson Da Costa/CEFET-MG

Coordenação do Curso
Adelson de Paula Silva/CEFET-MG

Professor-autor
Marcos Prado Amaral/CEFET-MG

Comissão de Acompanhamento e Validação
Universidade Federal de Santa Catarina – UFSC

Coordenação Institucional
Araci Hack Catapan/UFSC

Coordenação do Projeto
Silvia Modesto Nassar/UFSC

Coordenação de Design Instrucional
Beatriz Helena Dal Molin/UNIOESTE e UFSC

Coordenação de Design Gráfico
Juliana Tonietto/UFSC

Design Instrucional
Gustavo Pereira Mateus/UFSC

Web Master
Rafaela Lunardi Comarella/UFSC

Web Design
Beatriz Wilges/UFSC
Mônica Nassar Machuca/UFSC

Diagramação
Bárbara Zardo De Nardi/UFSC
Breno Takamine/UFSC
Liana Domeneghini Chiaradia/UFSC
Luiz Fernando Tomé/UFSC
Marília Ceriolli Hermoso/UFSC
Roberto Gava Colombo/UFSC

Revisão
Júlio César Ramos/UFSC

Projeto Gráfico
e-Tec/MEC

Catalogação na fonte elaborada pela DECTI da Biblioteca Central da UFSC

A485r Amaral, Marcos Prado
Redes de computadores II / Marcos Prado Amaral. – Belo Horizonte : CEFET-MG, 2012.
198p. : il. , tabs.

Inclui bibliografia
Curso técnico em planejamento e gestão em tecnologia da informação

ISBN: 978-85-99872-23-9

1. Redes de computadores. 2. TCP/IP (Protocolo de rede de computador). 3. Ensino a distância. I. Título. II. Título: Curso técnico em planejamento e gestão em tecnologia da informação.

CDU: 681.31.011.7

Apresentação e-Tec Brasil

Prezado estudante,

Bem-vindo a Rede e-Tec Brasil!

Você faz parte de uma rede nacional de ensino, que por sua vez constitui uma das ações do Pronatec - Programa Nacional de Acesso ao Ensino Técnico e Emprego. O Pronatec, instituído pela Lei nº 12.513/2011, tem como objetivo principal expandir, interiorizar e democratizar a oferta de cursos de Educação Profissional e Tecnológica (EPT) para a população brasileira propiciando caminho de o acesso mais rápido ao emprego.

É neste âmbito que as ações da Rede e-Tec Brasil promovem a parceria entre a Secretaria de Educação Profissional e Tecnológica (SETEC) e as instâncias promotoras de ensino técnico como os Institutos Federais, as Secretarias de Educação dos Estados, as Universidades, as Escolas e Colégios Tecnológicos e o Sistema S.

A educação a distância no nosso país, de dimensões continentais e grande diversidade regional e cultural, longe de distanciar, aproxima as pessoas ao garantir acesso à educação de qualidade, e promover o fortalecimento da formação de jovens moradores de regiões distantes, geograficamente ou economicamente, dos grandes centros.

A Rede e-Tec Brasil leva diversos cursos técnicos a todas as regiões do país, incentivando os estudantes a concluir o ensino médio e realizar uma formação e atualização contínuas. Os cursos são ofertados pelas instituições de educação profissional e o atendimento ao estudante é realizado tanto nas sedes das instituições quanto em suas unidades remotas, os polos.

Os parceiros da Rede e-Tec Brasil acreditam em uma educação profissional qualificada – integradora do ensino médio e educação técnica, - é capaz de promover o cidadão com capacidades para produzir, mas também com autonomia diante das diferentes dimensões da realidade: cultural, social, familiar, esportiva, política e ética.

Nós acreditamos em você!

Desejamos sucesso na sua formação profissional!

Ministério da Educação
Dezembro de 2012

Nosso contato
etecbrasil@mec.gov.br

Indicação de ícones

Os ícones são elementos gráficos utilizados para ampliar as formas de linguagem e facilitar a organização e a leitura hipertextual.

Atenção: indica pontos de maior relevância no texto.

Saiba mais: oferece novas informações que enriquecem o assunto ou "curiosidades" e notícias recentes relacionadas ao tema estudado.

Glossário: indica a definição de um termo, palavra ou expressão utilizada no texto.

Mídias integradas: sempre que se desejar que os estudantes desenvolvam atividades empregando diferentes mídias: vídeos, filmes, jornais, ambiente AVEA e outras.

Atividades de aprendizagem: apresenta atividades em diferentes níveis de aprendizagem para que o estudante possa realizá-las e conferir o seu domínio do tema estudado.

Sumário

Palavras do professor-autor

Caro estudante.

Parabéns pela sua escolha, pois o profissional de informática é hoje muito requisitado e tem um amplo campo de atuação. Um desses campos de atuação está intimamente ligado à internet, que é sua conectividade.

Na nossa disciplina de Redes de Computadores II (RCII) você aprenderá sobre as redes computacionais, protocolo TCP/IP e segurança de dados. O tema é grandioso e as perspectivas são muitas.

Nessa viagem, você terá a possibilidade de adquirir conhecimentos sobre como os dados trafegam pela internet, como manter uma rede conectada e a internet disponível e segura.

No intuito de facilitar o aprendizado, este caderno dispõe de figuras e vários exercícios ao final de cada aula. Esses exercícios têm a finalidade de fixar os conteúdos estudados. Procure fazê-los com base no conteúdo deste caderno e pesquisando na internet, em livros e artigos indicados.

Nunca fique com dúvidas e não deixe acumular o conteúdo apresentado. Estude um pouco todos os dias para que esse conteúdo se torne mais simples de assimilar.

Espero que você esteja motivado e curioso para que possa aproveitar ao máximo nossa viagem pelo mundo das redes computacionais.

Desejo sucesso no seu estudo e na sua carreira!

Marcos Prado Amaral

"O tempo é muito lento para os que esperam acontecer,
Muito rápido para os que têm medo de apreender,
Muito longo para os que lamentam o tempo perdido,
Muito curto para os que festejam as lições passageiras,
Mas, para os que amam o que fazem, o tempo é eterno."

Adaptado de William Shakespeare

Apresentação da disciplina

As redes computacionais adquiriram uma importância tão grande, que o seu bom funcionamento é imprescindível para as pessoas e, principalmente, para as empresas.

Por esse motivo, seu uso tem crescido a uma velocidade espantosa; conhecer o seu funcionamento e sua tecnologia pode ser um grande diferencial profissional na área de informática.

Considerando esse fato, esta disciplina levará você a ter contato com os principais conce itos relacionados a uma rede de computador baseada no protocolo TCP/IP.

Depois de estudar o conteúdo deste material, você terá compreensão da teoria e da prática do funcionamento de uma rede TCP/IP, assim como de seus principais componentes. Conhecerá e entenderá os aspectos envolvidos na configuração e manutenção de uma rede baseada no protocolo TCP/IP, bem como no gerenciamento e segurança desse tipo de redes de computadores.

Apresentação da disciplina

Disciplina: Redes de Computadores II (carga horária: 60h).).

Ementa: Apresentação da suíte TCP/IP; Comparação da suíte TCP/IP com o modelo OSI; As camadas da suíte TCP/IP; A camada de Enlace da suíte TCP/IP; A camada de internet da suíte TCP/IP; O protocolo IPv4; Endereçamento IPv4; Roteamento IPv4; Endereçamento CIDR; Protocolos de roteamento; A camada de transporte no modelo TCP/IP; As diferenças entre TCP e UDP; A camada de aplicação no modelo TCP/IP; O Sistema DNS; O protocolo DHCP; O protocolo NAT; , O PROXY; Correio eletrônico; Elementos ativos de rede: concentrador, pontes, roteadores, *gateways, hub, switch*; Gerência de redes; Segurança da informação; *Firewall*; Criptografia; O endereçamento IPv6; A relação do endereço IPv6 com o endereço IPv4.

AULA	OBJETIVOS DE APRENDIZAGEM	MATERIAIS	CARGA HORÁRIA (horas)
1. Arquitetura TCP/IP e Rede Integrada	Conhecer os protocolos e serviços que compõem o TCP/IP. Fazer um comparativo OSI X TCP/IP. Conhecer o protocolo Ethernet. Conhecer o surgimento da internet. Distinguir os serviços que trafegam pelas redes TCP/IP. Conhecer como os nomes na internet são definidos.	Vídeos instrucionais sobre Redes de Telecomunicações e Redes de Computadores. Ambiente virtual: http://etec.cefetmg.br/. Indicações de leitura; fórum de discussão; *chats*, e glossário.	10
2. Protocolos TCP/IP	Conhecer os protocolos que compõem a suíte TCP/IP. Conhecer os serviços WEB.		8
3. Endereçamento IPv4	Conhecer os endereços IPv4. Conhecer as classes e subclasses do endereço IPv4. Conhecer a máscara de sub-redes. Conhecer os endereços IPV6. Fazer uma comparação IPv4 *versus* IPV6.		14
4. Equipamentos básicos de conectividade	Conhecer o roteamento, seus algoritmos e protocolos. Conhecer os protocolos da camada de rede. Saber sobre os elementos ativos de uma rede de computadores.		14
5. Segurança da informação	Identificar os processos e sistemas de segurança dentro do contexto de uma visão ampla de políticas de segurança. Conhecer os risco e vulnerabilidades dos sistemas computacionais Conhecer as técnicas e mecanismos de defesa.		14

Aula 1 – Arquitetura TCP/IP e rede integrada

Objetivos

Conhecer os protocolos e serviços que compõem o TCP/IP.

Fazer um comparativo OSI X TCP/IP.

Conhecer o protocolo Ethernet.

Conhecer sobre o surgimento da internet.

Distinguir os serviços que trafegam pelas redes TCP/IP.

Conhecer como os nomes internet são definidos.

1.1 Modelo OSI

Devido ao surgimento de um grande número de redes de computadores a *International Organization for Standardization* (ISO), uma das principais organizações no que se refere à elaboração de padrões de âmbito mundial, criou, no início da década de 1980, um modelo de rede de sistema aberto para ajudar os desenvolvedores a implementar redes heterogêneas, que poderiam comunicar-se e trabalhar juntas (interoperabilidade) independentemente da tecnologia utilizada. Esse modelo, lançado em 1984, é chamado de modelo de referência OSI, que serve de base para implementação de qualquer tipo de rede.

Segundo Ulbrich e Valle (2003), pode-se dizer que o modelo de referência *Open System Interconnection* (OSI) nasceu da necessidade de criar um protocolo que conseguisse se comunicar entre redes diferentes.

No modelo de referência OSI existem sete níveis, ou camadas, numerados de 1 a 7 (vide Quadro 1.1) e cada um ilustra uma função particular da rede. De acordo com Soares (1995), um projeto de protocolo em níveis é a maneira mais eficiente de se estruturar uma rede, pois uma vez definida claramente a interface entre os diversos níveis, uma alteração na implementação de um nível pode ser realizada sem causar impacto na estrutura global. Assim, a

proposição do modelo OSI é "dividir para melhorar", fazendo com que o problema de transferir informações entre computadores seja dividido em sete problemas menores (camadas), de maneira que estes passem a ser mais fáceis de serem tratados e otimizados. Baseado nisso pode-se dizer que a divisão nessas sete camadas oferece as seguintes vantagens:

- Decompõe as comunicações de rede em partes menores e mais simples.
- Padroniza os componentes de rede, permitindo o desenvolvimento e o suporte por parte de vários fabricantes.
- Possibilita a comunicação entre tipos diferentes de *hardware* e de *software* de rede.
- Evita que as modificações em uma camada afetem as outras, possibilitando maior rapidez no seu desenvolvimento.
- Decompõe as comunicações de rede em partes menores, facilitando sua aprendizagem e compreensão.

Quadro 1.1: Camadas do modelo OSI

Nº da Camada	Nome da Camada	Função da Camada
7	Aplicação	Esta camada funciona como uma interface de ligação entre os processos de comunicação de rede e as aplicações utilizadas pelo usuário.
6	Apresentação	Aqui os dados são convertidos e garantidos em um formato universal.
5	Sessão	Estabelece e encerra os enlaces de comunicação.
4	Transporte	Efetua os processos de sequenciamento e, em alguns casos, confirmação de recebimento dos pacotes de dados.
3	Rede	O roteamento dos dados através da rede é implementado nesta camada.
2	Enlace	Aqui a informação é formatada em quadros (frames). Um quadro representa a exata estrutura dos dados fisicamente transmitidos através do fio ou outro meio.

Continua

1	Física	Define a conexão física entre o sistema computacional e a rede. Especifica o conector, a pinagem, níveis de tensão, dimensões físicas, características mecânicas e elétricas, etc.

Conclusão

Fonte: http://mesonpi.cat.cbpf.br/naj/tcpipf.pdf

A-Z

Pinagem
Descrição da posição de todos os pinos de um conector, juntamente com suas funções e sinais

Podemos identificar que a comunicação no modelo OSI se dá por camadas semelhantes, quando cada camada de um computador se comunica virtualmente com sua semelhante em outro computador. De acordo com Soares (1995), o processo de comunicação começa com a entrega dos dados a serem transmitido pelo usuário para uma entidade do nível de aplicação. Esses dados recebem o nome de Unidade de Dados do Serviço (SDU – *Service Data Unit*). A entidade que recebe os dados, nesse caso a camada de aplicação, junta aos dados um cabeçalho de controle chamado de Informação de Controle de Protocolo (PCI – *Protocol Control Information*). O objeto resultante dessa junção é chamado de Unidade de Dados do Protocolo (PDU – *Protocol Data Unit*). A PDU é a unidade de informação utilizada na troca virtual entre as entidades pares. Ela é passada para a camada imediatamente inferior. Nessa camada, essa PDU é a SDU dela, que adicionada ao seu PCI forma o PDU dessa camada. E assim ocorre sucessivamente até chegar à última camada, que é a camada física. Essa camada do destinatário está ligada fisicamente com a sua camada par do destinatário.

Ao chegar ao destinatário, ocorre o processo inverso, sendo que cada camada que recebe o PDU, retira o PCI que lhe corresponde, e passa para a camada imediatamente superior. Esse processo é repetido até a última camada que, ao retirar seu PCI, recupera o dado inicial enviado pelo remetente (vide Figura 1.1).

Figura 1.1: Esquema de comunicação no modelo OSI
Fonte: www-usr.inf.ufsm.br/~candia/aulas/espec/Aula_3_Modelo_OSI.pdf

A unidade de informação muda de nome ao longo das camadas, de maneira que podemos identificar, pelo nome dessas unidades, a qual camada se está

referindo. O Quadro 1.2 relaciona os diversos nomes dessas unidades de informação ao longo das camadas.

Quadro 1.2: Nomes das unidades de informação ao longo das camadas do modelo OSI

Nº da Camada	Nome da Camada	Unidade de Informação da Camada
7	Aplicação	Mensagem
4	Transporte	Segmento
3	Rede	Datagrama
2	Enlace	Frame/Quadro
1	Física	Bit

Reforce o aprendizado sobre o modelo RM-OSI assistindo ao vídeo disponível em: http://pontoderedes.blogspot.com/2010/02/video-aula-do-modelo-osi.html

Ao receber dados para efetuar um serviço, a camada N necessita incluir um cabeçalho, no qual são registradas informações relativas à camada. A esse cabeçalho damos o nome de Informação de Controle do Protocolo (PCI). Aos dados recebidos pela camada N, damos o nome de Unidade de dados do Serviço (SDU). Ao conjunto formado por PCI + SDU damos o nome de Unidade de Dados do Protocolo (PDU) (SENGER, 2012).

1.2 Suíte TCP/IP

O protocolo TCP/IP atualmente é o mais usado em redes locais. Isso se deve basicamente à popularização da internet, a rede mundial de computadores, já que esse protocolo foi criado para ser usado nela (TORRES, 2009).

Segundo Torres (2009), o TCP/IP é, na realidade, um conjunto de protocolos, dos quais os principais são justamente o Transmission Control Protocol (TCP) e o Internet Protocol (IP). A internet fornece vários serviços, que vão desde aplicações de baixo nível como o protocolo IP, passando pelos níveis de transporte como o TCP e UDP, até aplicações de alto nível como o HTTP, DNS, etc. Segundo Torres (2009), a arquitetura TCP/IP é composta por quatro camadas, que são: camada de Aplicação (camada 4), camada de Transporte (camada 3), camada de Internet ou de Rede(Camada 2) e camada Física ou Interface com a Rede (camada 1).

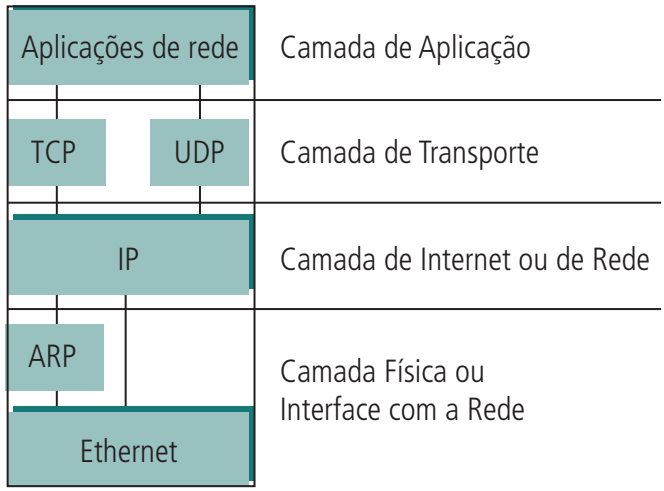


Figura 1.2: As camadas do protocolo TCP/IP
Fonte http://paginas.fe.up.pt/~goii2000/M3/tcpip2.htm

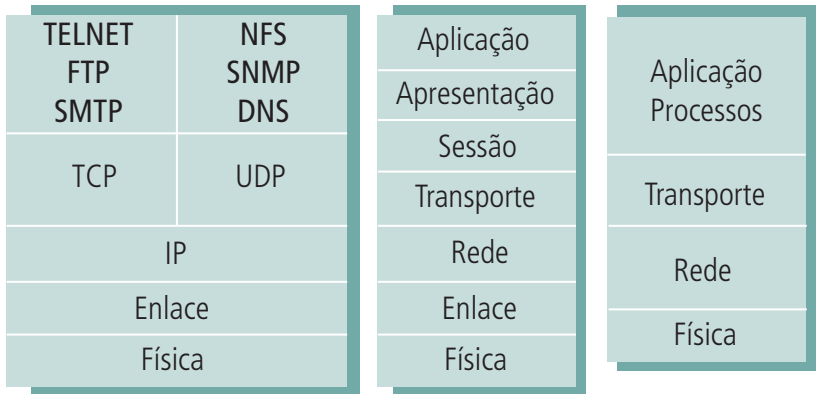


Figura 1.3: Relação das camadas do modelo OSI com as do protocolo TCP/IP
Fonte: http://mesonpi.cat.cbpf.br/naj/tcpipf.pdf

Na comparação da suíte TCP/IP com o modelo OSI, pode-se notar que as camadas da suíte TCP/IP têm o mesmo nome de algumas camadas no modelo OSI (vide Figura 1.3). Mas, apesar disso, elas não são iguais, pois têm funções e protocolos característicos.

Podemos dizer que as quatro camadas do protocolo TCP englobam todos os serviços das sete camadas do modelo OSI.

A **primeira camada** da suíte TCP/IP não é definida, pois isso faz com que o TCP/IP não dependa do meio físico para operar. Mesmo assim, ela tem importância vital em relação à interligação física na comunicação, manipulando os *bits*, os quadros de *bits*, os endereços MAC, fazendo a checagem de erro (engloba a camada de enlace e física do modelo ISO/OSI), etc. É da camada física a função de conectar o TCP/IP com os diversos tipos de redes (X.25, ATM, FDDI, Ethernet, Token Ring, Frame Relay, sistema de conexão ponto a ponto SLIP, etc.).

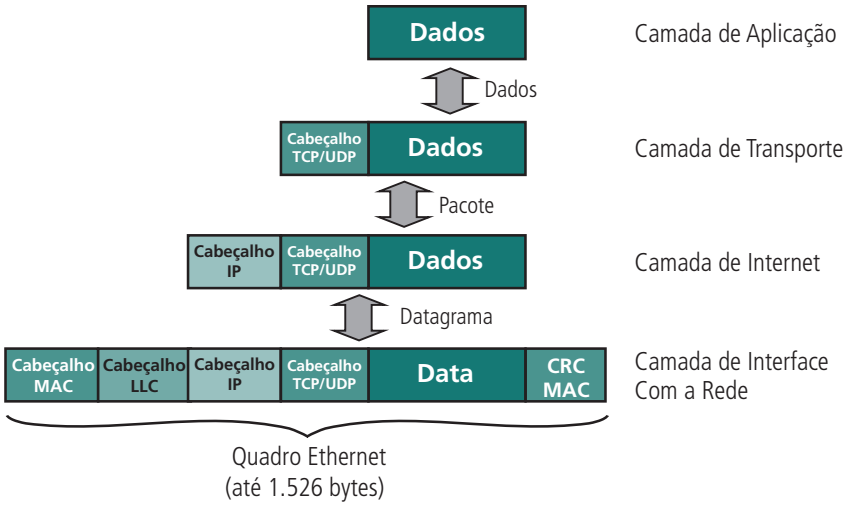


Figura 1.4: Quadro da camada de Interface com a Rede
Fonte: http://www.clubedohardware.com.br/artigos/Como-o-Protocolo-TCP-IP-Funciona-Parte-1/1351/6

A **segunda camada**, o protocolo IP, é responsável pela conexão entre os sistemas que estão se comunicando, pavimentando o caminho dos dados pela rede. É nesta camada/protocolo que a informação é fragmentada no sistema fonte e reagrupada no sistema alvo. Cada um desses fragmentos pode ter caminhos diferentes pela rede, de forma que os fragmentos podem chegar fora de ordem. Ela é usada para atribuir endereço de rede (IP) ao sistema e rotear a informação para a rede correta, ou seja, descobrir o caminho ou rota por onde a informação deverá passar. Por esse motivo, a camada IP é não orientada à conexão, comunicando-se por datagrama. No modelo ISO/OSI, a **camada 3** é a equivalente ao protocolo IP.

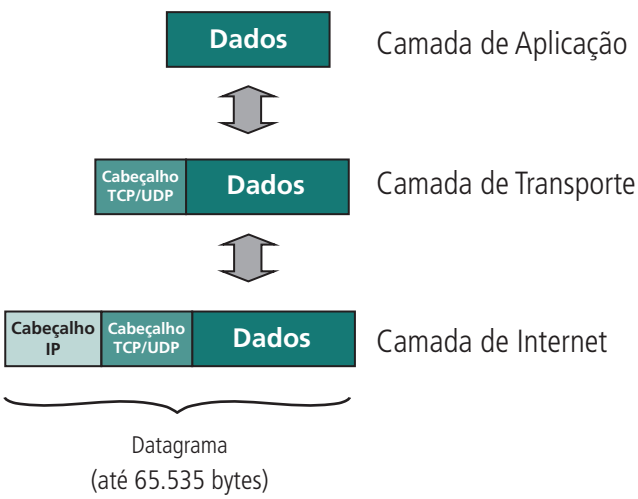


Figura 1.5: Datagrama da camada de Internet
Fonte: http://www.clubedohardware.com.br/artigos/Como-o-Protocolo-TCP-IP-Funciona-Parte-1/1351/6

A **terceira camada** de transporte é responsável pela liberação dos dados para o destino. Ela possui dois protocolos: TCP e UDP. Ela tem a função principal de começar e terminar uma conexão e ainda controlar o fluxo de dados e de efetuar processos de correção e verificação de erros.

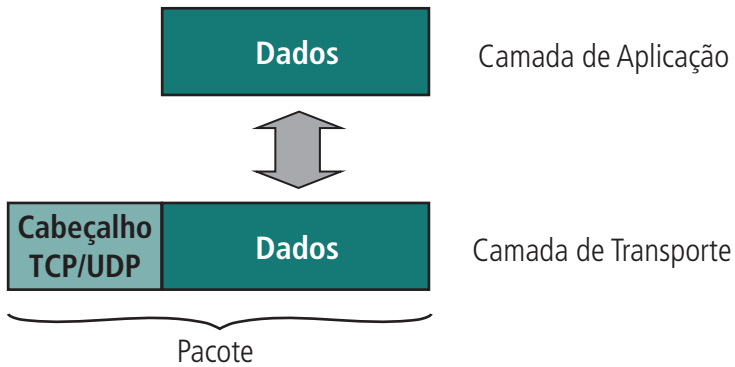


Figura 1.6: Pacote de dado na camada de Transporte
Fonte: http://www.clubedohardware.com.br/artigos/Como-o-Protocolo-TCP-IP-Funciona-Parte-1/1351/6

A **quarta camada** é composta pelos protocolos de processos da Internet. Os processos possuem o nome do próprio protocolo utilizado. É nessa camada que se estabelece o tratamento das diferenças entre representação de formato de dados. O endereçamento da aplicação na rede é provido através da utilização de portas para comunicação com a camada de transporte. Muitas vezes esses protocolos são chamados de cliente-servidor. A suíte TCP/IP não faz distinção entre as camadas superiores como no modelo ISO/OSI (**camadas 5, 6 e 7**).

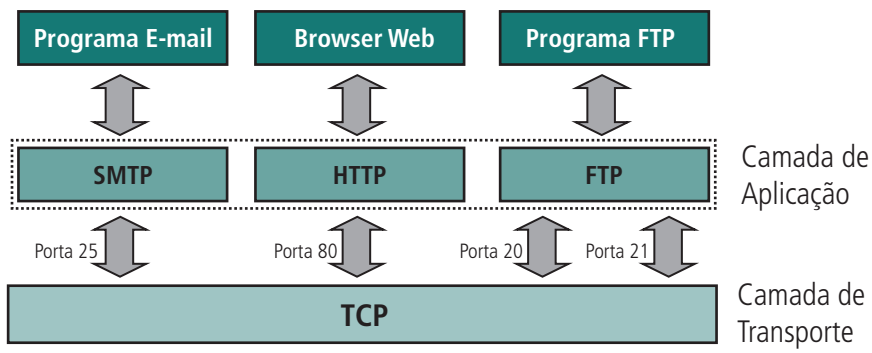


Figura 1.7: Como a camada de aplicação funciona
Fonte: http://www.clubedohardware.com.br/artigos/Como-o-Protocolo-TCP-IP-Funciona-Parte-1/1351/6

1.3 Comunicação TCP/IP

O protocolo TCP/IP trabalha em redes comutadas por pacotes, ou seja, ele quebra os dados em seções chamadas pacotes (responsabilidade do TCP) e entrega esses pacotes em seu destino (responsabilidade do IP). Na recepção dos dados, o TCP/IP remonta os pacotes na sua forma original. A arquitetura

do TCP/IP baseia-se principalmente em um serviço de transporte orientado à conexão fornecido pelo TCP, e em um serviço de rede não orientado à conexão (datagrama não confiável) fornecido pelo IP. As especificações de cada protocolo contidas na suíte de protocolos do TCP/IP estão definidas em um documento chamado *Request for Comments* (RFC), que pode ser obtido em **http://www.faqs.org/rfcs/**.

Reforce o aprendizado sobre *Request for Comments* (RFC) acessando o *site* http://www.faqs.org/rfcs/

O tráfego na rede, de acordo com o modelo TCP/IP, se organiza na seguinte ordem:

1. Um processo ou aplicação gera um DADO e o envia à camada *Host-to-Host*.
2. Na camada *Host-to-Host*, quando adicionado um cabeçalho TCP ao DADO, ele é denominado MENSAGEM.
3. Na camada INTERNET, quando adicionado um cabeçalho IP à MENSAGEM, ela passa a ser denominada PACOTE.
4. Na camada de ACESSO À REDE, quando adicionado um cabeçalho com o endereço MAC ou físico da estação ao PACOTE e alguns dígitos de controle, é denominado *Frame*;
5. O Frame é transportado ao seu destino.
6. Chegando ao seu destino, é feito o processo inverso de desmontagem do *Frame*, Pacote, Mensagem e entregue o dado ao processo ou aplicação de destino.

1.4 Modelo OSI x TCP/IP

1.4.1 Algumas semelhanças

- Ambos são divididos em camadas.
- Ambos têm camadas de aplicação, embora incluam serviços muito diferentes.
- Ambos têm camadas de transporte e de rede comparáveis.
- A tecnologia de comutação de pacotes (e não comutação de circuitos) é presumida por ambos.
- Os profissionais da rede precisam conhecer ambos.

1.4.2 Algumas diferenças

- O TCP/IP combina os aspectos das camadas de apresentação e de sessão dentro da sua camada de aplicação.
- O TCP/IP combina a camada física e enlace do OSI em uma camada.
- O TCP/IP parece ser mais simples por ter menos camadas.
- Os protocolos do TCP/IP são os padrões em torno dos quais a Internet se desenvolveu; portanto o modelo TCP/IP ganha credibilidade apenas por causa dos seus protocolos.
- Em contraste, nenhuma rede foi criada em torno de protocolos específicos relacionados ao OSI, embora todos usem o modelo OSI para guiar os estudos.

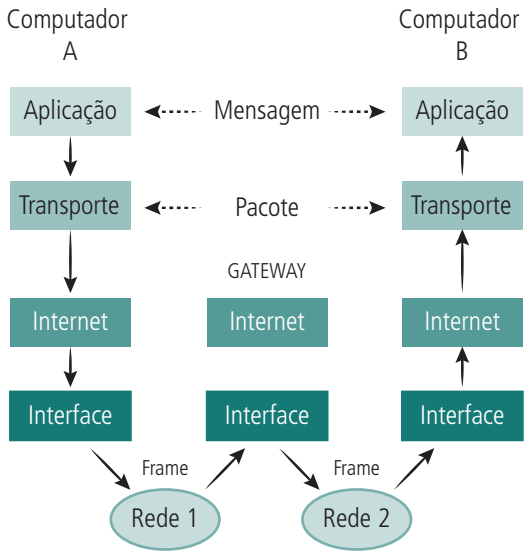


Figura 1.8: Fluxo de mensagens
Fonte: http://www.ccuec.unicamp.br/treinamento_int2004/tcpip/05.html

1.5 Protocolo Ethernet

Segundo Bernal (2012), o protocolo Ethernet faz parte da camada de enlace (**camada 2** do modelo OSI). Essa camada não é descrita pela suíte TCP/IP (Faz parte da **camada 1**), mas é um protocolo importante na comunicação entre redes, devido à abrangência de seu uso em redes locais (LAN). Esse protocolo é o método usado para passar pacotes da camada de rede de um dispositivo para a camada de internet de outro em uma LAN. Esse processo pode ser controlado tanto em *software* (*device driver*) para a placa de rede quanto em *firmware* ou *chipsets* especializados.

1.5.1 Redes locais LAN

Como visto anteriormente, uma LAN é uma rede de computadores desenvolvida para cobrir regiões geograficamente pequenas; nesse universo de

LANs espalhadas pelo mundo, o padrão mais utilizado em nível de enlace é o protocolo Ethernet.

1.5.2 Camada de enlace do modelo OSI

A camada de enlace tem a função de executar seus serviços nos dados a serem transmitidos e adicionar um *header* de pacote para prepará-lo para transmissão (como visto no modelo OSI) e transmitir o quadro através da camada física. No receptor, a camada de enlace irá receber quadros de dados, retirar os *headers* adicionados e encaminhar os pacotes recebidos para a camada de rede ou internet. Essa camada é a primeira normatizada do modelo OSI, e é responsável pelo controle do endereçamento físico ou MAC e controle de envio e recepção dos dados em forma de quadros. Ela não é orientada à conexão, comunica-se pelos datagramas (pacotes de dados).

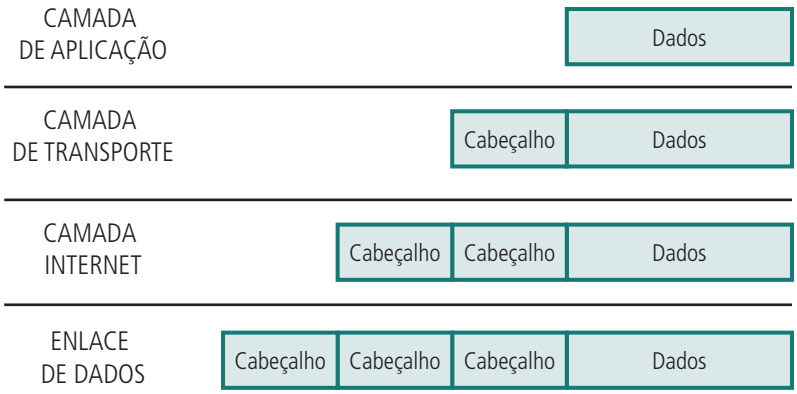


Figura 1.9: Encapsulamento TCP/IP-Ethernet
Fonte: http://www.ccuec.unicamp.br/treinamento_int2004/tcpip/05.html

1.5.3 O quadro Ethernet

Como o protocolo Ethernet trabalha na camada de enlace, segundo Comer (2007), faz sentido observar que suas informações são transmitidas como um quadro entre computadores, através da rede. Esse quadro possui um cabeçalho informando a origem, o destino, quais dados estão sendo transmitidos, além de outras informações úteis. O formato do quadro é padronizado, para que computadores diferentes possam se comunicar.

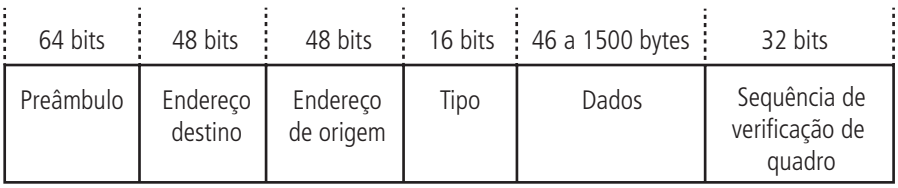


Figura 1.10: Quadro Ethernet
Fonte: http://carlosalex.ueuo.com/carlos/disciplinas/rc/Capitulo_01_Ethernet.pdf

Segundo Tanenbaum (2003), o quadro Ethernet é dividido em campos (vide Figura 1.10), que podem ser descritos da seguinte maneira:

- Preâmbulo: tem como função criar um padrão de 0s e 1s para sincronização.
- Endereço de destino: contém o endereço físico (MAC) da estação de destino.
- Endereço de origem: contém o endereço físico (MAC) da estação de origem.
- Tipo: identifica o tipo de dados do *payload*.
- *Payload* (dados): transporta os dados encapsulados pelos protocolos das camadas superiores (entre 46 e 1.500 *bytes*). Se a parte de dados de um quadro for menor que 46 *bytes*, um campo Preenchimento será usado para preencher o quadro até o tamanho mínimo.
- CRC (Sequência de Verificação de Quadro): transporta a informação da detecção de erro (CRC-32).

1.5.4 Subcamadas Ethernet

Segundo Bernal (2012), alguns anos após a definição do Ethernet, o Instituto dos Engenheiros Eletrônicos e Eletricistas (IEEE) definiu o padrão IEEE 802.2, semelhante ao padrão Ethernet, e dividiu-o em três subcamadas, que são a subcamada LLC (*Logical Link Control* ou Controle Lógico do *Link*), a subcamada MAC (*Medium Access Control* ou Controle de Acesso ao Meio) e a subcamada Física.

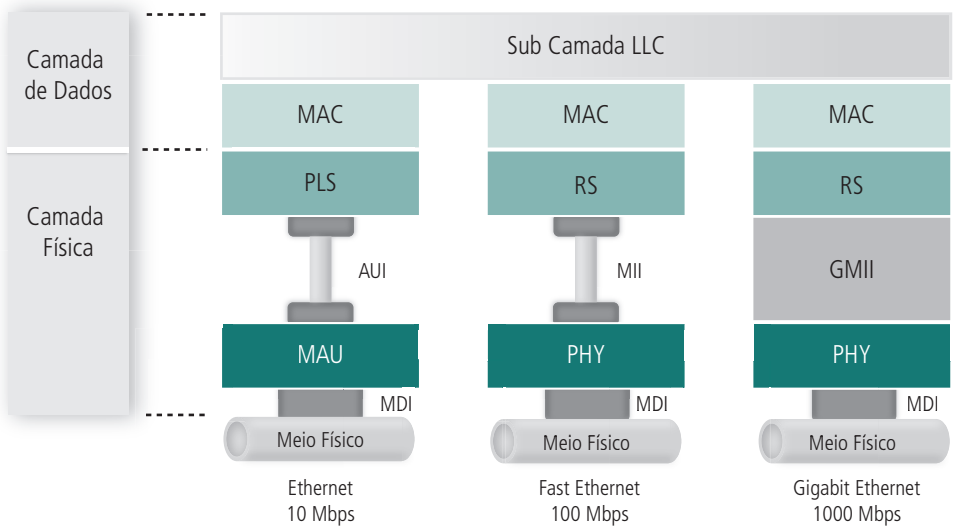


Figura 1.11: Comparativo de três gerações da Ethernet
Fonte: http://www.ifba.edu.br/professores/romildo/downloads/ifba/ethernet.pdf

1.5.5 Subcamada MAC

Esta subcamada controla a operação de acesso ao meio físico, além de receber *frames* da camada superior e enviá-los à camada inferior para codificação. Para efetuar tal controle de acesso, é utilizado o método CSMA/CD.

No protocolo CSMA/CD, Quando um computador deseja enviar alguma informação, este obedece aos seguintes algoritmos:

1. Se o canal está livre, inicia-se a transmissão; senão, vai para o passo 4.

2. Na transmissão da informação, se colisão é detectada, a transmissão continua até que o tempo mínimo para o pacote seja alcançado (para garantir que todos os outros transmissores e receptores detectem a colisão); então segue para o passo 4.

3. No fim de transmissão com sucesso, é informado o sucesso para as camadas de rede superiores, e o sistema sai do modo de transmissão.

4. Se canal está ocupado, espera-se até que o canal esteja livre.

5. Quando o canal se torna livre, espera-se um tempo aleatório1, e vai-se para o passo 1, a menos que o número máximo de tentativa de transmissão tenha sido excedido.

6. Quando o número máximo de tentativa de transmissão for excedido, informa-se a falha para as camadas de rede superiores, sai-se do modo de transmissão.

Através da subcamada MAC, o protocolo Ethernet oferece a comunicação entre equipamentos de uma mesma rede física sem uso de conexões e com serviços ***unicast*** (um quadro vai para um destino único), ***multicast*** (um quadro vai para múltiplos destinos) e **difusão** (um quadro vai para todos os destinos).

Mas para haver essa comunicação, cada equipamento participante deve possuir um endereço que é chamado de endereço MAC (*Media Access Control*). Os endereços MAC possuem 48 *bits* e são únicos por construção; isso significa que quando um fabricante constrói uma placa de rede Ethernet, ela recebe um endereço único determinado por *hardware* e esse endereço é único no mundo inteiro (espaço plano ou *flat address space*).

1.5.6 Endereçamento MAC

Cada estação numa rede Ethernet possui seu próprio adaptador de rede que contém um identificador MAC.

Seja exemplo de uma placa de rede qualquer com as seguintes descrições:

Descrição . . . . . : VIA *Rhine* II *Fast Ethernet Adapter*

Endereço físico . . . . . : 00-1A-4D-A4-6A-E5

Nesse exemplo, podemos ver que no endereço físico, ou MAC *address*, os três primeiros octetos são destinados à identificação do fabricante, os três posteriores são fornecidos pelo fabricante. É um endereço universal, por isso não existem, em todo o mundo, duas placas com o mesmo endereço.

Relação MAC x Empresas

00-00-0C (hex) CISCO SYSTEMS INC
00-01-41 (hex) CISCO SYSTEMS INC
00-00-63 (hex) CISCO SYSTEMS INC
00-01-02 (hex) 3COM Corporation
00-01-03 (hex) 3COM Corporation
00-E0-4C (hex) REALTEK CORP.

Observe que uma empresa pode possuir mais que um grupo de MAC não contínuo.

Figura 1.12: Relação endereço MAC x Fabricante placa de rede
Fonte: http://www.ifba.edu.br/professores/romildo/downloads/ifba/ethernet.pdf

O endereço MAC é muito importante porque a comunicação em uma rede LAN com protocolo TCP/IP é feita por ele. Isso quer dizer que o endereço conhecido inicialmente pela estação transmissora é o endereço IP da estação destino, mas não seu endereço MAC. O mapeamento de endereços IP para endereços MAC é feito com o protocolo ARP (*Address Resolution Protocol*).

A Ethernet permite que quadros sejam enviados para endereços especiais. O endereço FF-FF-FF-FF-FF-FF é o endereço de difusão: o quadro enviado para tal endereço é recebido por todas as estações. Além do mais, cada interface de rede (placa de rede) pode ser configurada para receber quadros pertencentes a um grupo *multicast*. Os endereços *multicast* iniciam com um *bit* igual a 1.

1.5.7 Subcamada LLC

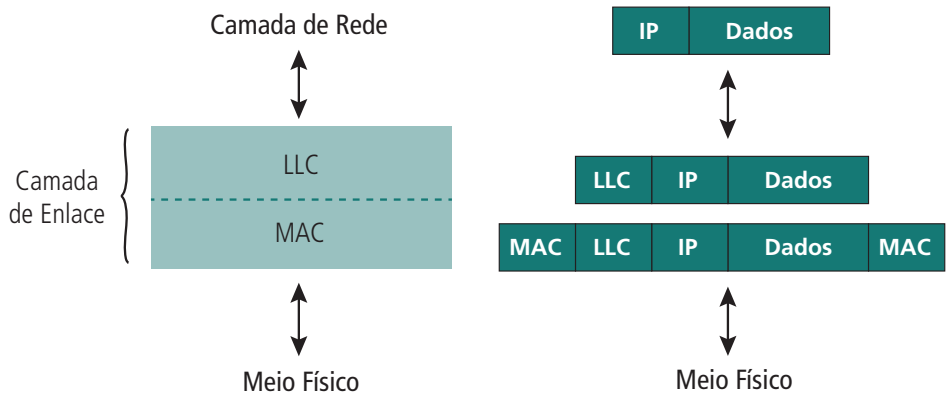


Figura 1.13: Subcamada LLC
Fonte: Elaborada pelo autor

A subcamada LLC (*Logic Link Control*) é a mais alta da Camada de Enlace. Ela fornece mecanismos de multiplexação e controle de fluxo que torna possível para os vários protocolos de rede trabalharem juntos dentro de uma rede multiponto e serem transportados pelo mesmo meio da rede.

Segundo Tanenbaum (2003), O LLC especifica os mecanismos para endereçamento de estações conectadas ao meio e para controlar a troca de dados entre utilizadores da rede. Também estabelece três tipos de serviço:

- Serviço de datagrama não confiável.
- Serviço de datagrama com confirmação.
- Serviço confiável orientado a conexões.

Pela Figura 1.14 podemos ver um quadro LLC, onde além da informação podemos definir três campos, que são:

- DSAP ou Ponto de acesso de destino: especifica o número de ponto de acesso de serviço do destino.
- SSAP ou Ponto de acesso de origem: especifica o número de ponto de acesso de serviço da fonte.
- CTRL ou Campo de controle: indica ações que devem ser tomadas de acordo com o tipo de conexão.

MSDU:

D S A P	s s A P	CONTROL	INFORMAÇÃO
8 Bits	8 Bits	8 Bits	M *8 Bits

DSAP - **identifica o endereço do ponto de acesso ao serviço de LLC na entidade destinatária**

SSAP - **identifica o endereço do ponto de acesso ao serviço LLC na entidade originadora**

CONTROL - **Campo de controle, identifica o formato de informação, supervisor ou não-numerado e suas peculiaridades**

INFORMAÇÃO - **Informações dos usuários que podem ou não estar presentes num limite determinado pelo método de acesso usado em particular**

Figura 1.14: Quadro da subcamada LLC
Fonte: http://penta2.ufrgs.br/tp951/protocolos/13llcb.html

1.6 Camada física

Existem diferentes tipos de implementação da camada física, que variam de acordo com a velocidade, topologia utilizada, padrões de cabeamento, conectores, entre outros. As categorizações de acordo com a velocidade são:

- *Standard Ethernet* – 10Mbps: Padrões: 10BASE2 / 10BASE5 / 10BASE--T / 10BASE-F (10BASE-FL, 10BASE-FB e 10BASE-FP) 10BASE2 (também chamado ThinNet ou Cheapernet) – um cabo coaxial de 50-ohm conecta as máquinas, cada qual usando um adaptador T para conectar seu NIC. Requer terminadores nos finais.
- *Fast Ethernet* – 100Mbps: padrões: 100BASE-T (100BASE-TX, 100BASE--T4 e 100BASE-T2) e 100BASE-FX.
- *GigaBit Ethernet*: Padrões: 1000BASE-T / 1000BASE-SX / 1000BASE-LX / 1000BASE-CX.
- 10 *GigaBit Ethernet*: Padrões: 10GBASE-SR/10GBASE-LX4/10GBASE--LR/10GBASE-ER/10GBASE-SW/10GBASE-LW/10GBASE-EW.

1.7 Internet

A internet tem sido aclamada como a mais revolucionária tecnologia que a computação já implementou. É uma tecnologia que está inserida no nosso dia a dias e não apenas no mundo da computação: hoje não se consegue ligar a televisão sem ver *flashes* de endereços da internet aparecendo na tela, ou ler um jornal sem ver uma história sobre a última atualidade na internet.

A internet é composta por inúmeros serviços como: apresentação de páginas eletrônicas nas quais é possível: preencher formulários de cadastro, efetuar compras e pesquisa de documentos relacionados a um determinado assunto no mundo inteiro, se necessário; conversar utilizando um telefone ou videoconferência sem mesmo ter que se preocupar em pagar uma ligação interurbana (o custo é apenas da ligação local até o provedor de serviços de internet (ISP) e o custo da sua conta de internet nesse provedor); além disso, pode-se comunicar de forma mais simples como o correio eletrônico, que possibilita trocar mensagens com todos que possuam uma conta na internet.

Quando a tecnologia da internet é aplicada e usada dentro de uma corporação, e é aberta apenas a seus empregados, ela é referida como uma intranet. A intranet usa as mesmas tecnologias que compõem a internet, com a única diferença em que a corporação isola-a da internet para manter fora os intrusos. Esse muro é conhecido como bloqueio ou *firewall*.

Quando uma parte dessa intranet é estendida a usuários externos específicos, tais como representantes e clientes, ela é chamada de extranet. Assim, o uso da extranet ocorre quando a liberação da "parte privada" de um *site* na qual somente "usuários registrados" podem navegar, previamente autenticados por sua senha (*login*).

1.7.1 Origem da internet

As redes que usam comutação de pacotes (a internet é uma delas) desenvolveram-se com o apoio e financiamento do governo dos EUA. A Advanced Research Projects Agency (ARPA) foi quem adotou inicialmente a teoria de comutação de pacotes.

Os esforços iniciais da ARPA, com a colaboração de várias empresas e universidades, segundo Tanenbaum (2003), levaram a uma rede experimental com quatro nós, que entrou em operação em dezembro de 1969. Logo esses nós passaram a trocar pacotes por meio de linhas telefônicas. Durante uma década, a ARPAnet cresceu em um ritmo de aproximadamente um novo computador a cada três semanas.

Segundo Carvalho (2012), em 1987 a Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) e o Laboratório Nacional de Computação Científica (LNCC) se conectaram a instituições dos EUA. Após conseguirem acesso a redes internacionais, essas instituições incentivaram outras entidades do Brasil a usar essas redes. Em 1988 a UFRJ conectou-se à UCLA. Em seguida, várias universidades e centros de pesquisa conectaram seus equipamentos a uma dessas instituições. Nasce, então, a internet no Brasil.

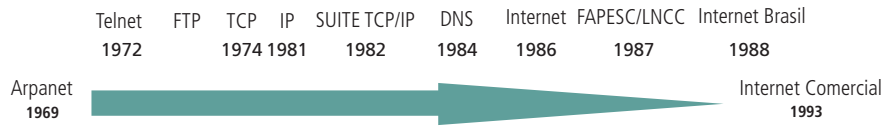


Figura 1.15: Evolução da internet
Fonte: Elaborada pelo autor

1.7.2 Administração da internet

Tanto a administração quanto a operação da internet são descentralizadas; apenas alguns serviços, tais como definição de padrões e pesquisas e ainda a distribuição dos endereços, são administrados por instituições regulamentadoras. As principais instituições são, segundo Valle (2012):

- A ISOC (*Internet Society*) – procura orientar a pesquisa e utilização através de fóruns, debates e publicações.
- O IAB (*Internet Architeture Board*) – fundado em 1983 e integrada ao ISOC em 1992, coordena os grupos IETF e IRTF descritos abaixo, na pesquisa e desenvolvimento envolvidos no funcionamento da internet.
- A IRTF (*Internet Research Task Force*) – grupo de pesquisadores que se dedicam a projetos de longo prazo referentes ao funcionamento da internet.
- A IETF (*Internet Engineering Task Force*) – grupo de pesquisadores responsáveis pelo desenvolvimento que no princípio tinham a intenção de apresentar propostas que se tornassem padrões oficiais da internet.
- O InterNIC (*Internet Network Information Center*) – composto de três instituições (AT&T, PSI e General Atomics) que organizam a distribuição dos endereços e registros de domínios e também das RFCs.
- A IANA (*Internet Assigned Numbers Authority*) – mantida pelo Instituto de Ciência e Informação da Universidade do Sul da Califórnia, controla a distribuição dos identificadores para serviços a serem oferecidos pela internet.

Ainda de acordo com Valle (2012), no Brasil, o principal órgão de administração da rede é o Comitê Gestor Internet, criado em 1995 por iniciativa do Ministério das Comunicações e do Ministério da Ciência e Tecnologia e seu principal objetivo é coordenar o acesso à internet no Brasil. A Rede Nacional de Pesquisas (RNP) administra *backbone* internet do Brasil.

1.7.3 Acessos à internet

No que se refere à execução de tarefas, Valle (2012) classifica o acesso de um computador à internet em:

- **Acesso completo**: o computador possui *software* TCP/IP (é endereçável na internet) e pode executar aplicações que podem interagir diretamente com outras aplicações residentes em outros computadores da internet; o computador é, portanto, um '*host*' da internet.

- **Acesso limitado**: o computador não possui *software* TCP/IP; apenas conecta-se a um computador que possui acesso completo à internet (por exemplo, conecta-se a esse computador via um emulador de terminal), ou seja, seu acesso à internet é indireto, através de programas residentes nesse computador; nesse caso o computador não é um '*host*' da internet.

Segundo Valle (2012), existe também outra classificação quanto à forma com que se dá a conexão entre um computador (ou uma rede de computadores) ao seu ponto de acesso à internet, que pode ser:

- **Conexão permanente**: a ligação entre os computadores e a internet é feita através de circuitos dedicados de comunicação. Esse tipo de ligação é usado por computadores que possuem acesso completo à internet, endereço fixo e nome de domínio fixo, e são localizáveis por qualquer outro computador em mesma situação.

- **Conexão temporária**: esse tipo de ligação é usado tanto por computadores com acesso completo quanto limitado à internet, e é feito normalmente através de linhas telefônicas discadas (o acesso à internet só existe enquanto a ligação telefônica está estabelecida). Nesse caso eles não são mais localizáveis de forma unívoca, pois normalmente não possuem um endereço fixo nem nome de domínio próprio.

Segundo Valle (2012) a integração entre essas classificações destacadas anteriormente assume a forma de acesso à internet conhecido por:

- **Acesso dedicado**: via conexão permanente, com acesso completo à internet, execução de aplicações clientes e servidoras.
- **Acesso discado de protocolo**: via conexão temporária, com acesso completo à internet, execução apenas de aplicações clientes.
- **Acesso discado de terminal**: via conexão temporária, com acesso limitado á internet, via emulação de terminal e/ou transferência de informações via protocolos não TCP/IP.

1.7.4 Utilização da internet

Em função do objetivo da conexão à internet, os usuários e instituições conectados podem ser classificados como provedores de serviços de internet (ISP). Os provedores de serviços internet são instituições conectadas à internet com o objetivo de fornecer serviços a ela relacionados. Não há consenso a respeito dessa classificação, pois em muitos casos é difícil enquadrar uma instituição em apenas uma delas. Nos Estados Unidos, por exemplo, o termo ISP (*Internet Service Provider*) é usado de forma geral para denominar o que aqui classificamos como provedores de acesso, podendo, em alguns casos, ser usado também para provedores que se aproximam em porte aos classificados como provedor de *backbone*. Verificaremos que os serviços fornecidos, pelos provedores podem ser classificados de várias formas, que, segundo Kuwabara (2012), são:

- **Provedores de backbone internet**: são instituições que constroem e administram *backbones* de longo alcance, com o objetivo de fornecer acesso à internet para redes locais, através de pontos de presença; a RNP é um exemplo desse tipo de provedor, com seu *backbone* internet/BR.
- **Provedores de acesso internet**: são instituições que se conectam à internet por meio de acessos dedicados, através de um provedor de *backbone*, para disponibilizar acessos a terceiros a partir de suas instalações.
- **Provedores de informação internet**: são instituições que disponibilizam informações através da internet. O acesso às informações é disponibilizado através de programas servidores como FTP, Gopher e www. Essas informações podem estar organizadas em bases de dados locais ou distribuídas pela internet.

1.7.5 Usuários da internet

Para Valle (2012), os usuários da internet podem ser classificados em:

- **Usuários individuais**: são, em geral, pessoas físicas que se conectam à internet com objetivos vários, desde o de utilizar recursos de correio eletrônico até o de divulgação de serviços pessoais. Normalmente seu acesso é do tipo discado, entre seu computador pessoal e as instalações de um provedor de acesso.
- **Usuários institucionais**: são empresas que conectam a sua rede corporativa, ou parte dela, à internet com o objetivo de fornecer acesso para seus funcionários, utilizando-a como "meio de comunicação" entre filiais e clientes, ou mesmo para praticar comércio. O seu acesso pode ser desde um tipo discado de protocolo envolvendo apenas um único equipamento da empresa, até um do tipo dedicado conectando toda a sua rede corporativa à internet. Esse acesso é normalmente obtido via provedor de acesso.

Aprenda sobre as conexões dos provedores de internet, acessando:

http://www.youtube.com/watch?v=NlhuSqqC1xs&feature=related

http://www.youtube.com/watch?v=VAeuOvJ3Ang&feature=related

1.7.6 Páginas na internet

Talvez o serviço mais conhecido da internet, e que fez com ela tenha a penetração de hoje, é o WWW (*World Wide Web*), ou ainda, a "grande teia", que é conhecido como página na internet, que tem a facilidade de incorporar varias ferramentas gráficas. Isso torna fácil a publicação e difusão das informações, pois permite às pessoas incorporarem textos, gráficos, sons, animações e outros elementos de multimídia, que levam essas páginas a terem uma linguagem mais universal. Em sua essência, cada página é uma publicação interativa de multimídia. Isso significa que uma companhia pode facilmente publicar desde documentos simples até páginas sofisticadas, que permitem às pessoas ler relatórios, ou ver vídeos ou mesmo participar de palestras *on-line* via computadores. A página de início (ou entrada do local) de uma organização é chamada de *home page*.

1.7.7 *Browser*

Para se comunicarem com um servidor *web*, os usuários finais rodam programas-clientes ou *browsers*, assim chamados porque pedem o serviço ao servidor. No caso da internet, o *browser* nada mais é do que uma interface que facilita a visualização dos serviços que o servidor que está sendo acessado pode disponibilizar. Os *browsers* acessam *home pages* ou páginas *web* que disponibilizam serviços como *e-mail*, busca de documentos, acesso a arquivos e muito mais. A internet é uma tecnologia aberta e, por sua vez, padronizada. Portanto, todos os serviços disponibilizados na rede são compatíveis com os

browsers existentes; porém, em alguns *sites* WWW (servidores da internet) poderá ser necessária a utilização de *plug-ins* que serão adicionados ao *browser* para facilitar a visualização de imagens 3D, sons, filmes e até mesmo a comunicação através de sons, como se fossem um telefone ou videoconferência.

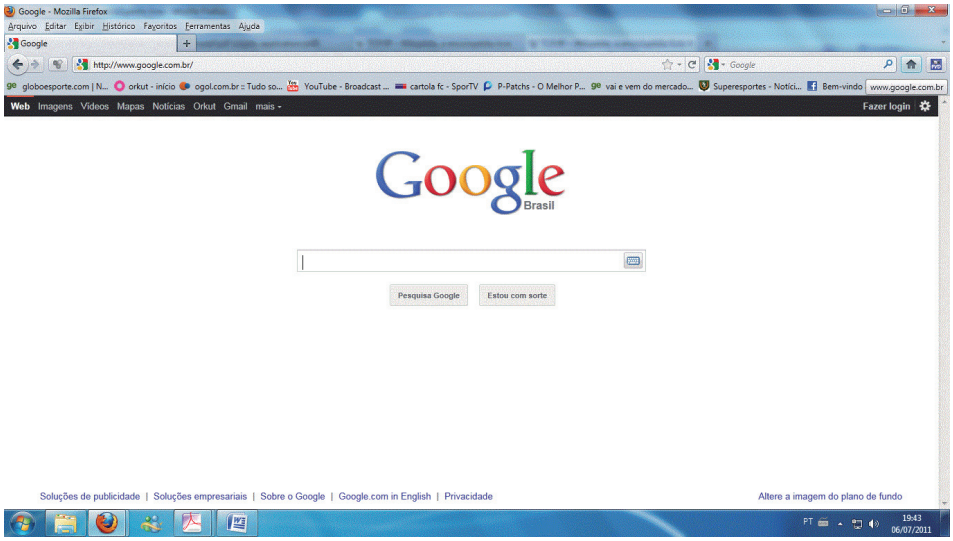


Figura 1.16: *Browser* cliente Firefox carregado com uma página *web*
Fonte: *Print screen* Mozila Firefox, 2012

1.7.8 Domínio internet

Domínio é um nome que serve para localizar e identificar conjuntos de computadores na internet. O nome de domínio foi concebido com o objetivo de facilitar a memorização dos endereços de computadores na internet. É a partir dele que os sítios são acessados, pois é mais fácil a memorização dos endereços "www" do que uma sequência grande de números, que é o endereço IP (BRASIL, 2012).

Essa atribuição de domínios na internet teve como objetivo evitar que dois equipamentos utilizassem o mesmo nome e também serviu para descentralizar o cadastramento de redes e equipamentos, dividindo-se a internet nesses domínios administrativos. Dessa maneira, o nome simbólico do um equipamento é composto por um termo local associado à hierarquia de domínios. Esse nome é conhecido como FQDN (*Fully Qualified Domain Name*).

Uma entidade poderá registrar, sob uma extensão, quantos domínios quiser. Porém, não é permitido registrar o mesmo nome em diferentes DPNs (Domínio de Primeiro Nome) genéricos. A restrição de homonímia não se aplica às extensões com restrições. Todas as extensões de domínio disponíveis, excetuando-se os restritos, são classificadas como genéricas.

Vejamos um exemplo: uma entidade poderá registrar quantos domínios quiser sob **com.br**, ou sob **ind.br**, mas, se possuir o domínio **xxx.com.br**, não poderá registrá-lo também em **ind.br**. Ou seja, se tiver **xxx.com.br** não poderá registrar **xxx.ind.br**, por se tratar de domínios genéricos. Nada impede que, caso essa entidade preencha os requisitos para registrar sob **tv.br**, registre também o **xxx.tv.br**, porque **tv.br** é um domínio com restrições próprias, às quais não se adicionam as restrições de homonímia (REGISTROBR, 2012).

A extensão **nom.br** é uma exceção à regra da homonímia. Por exemplo: pessoas físicas podem registrar **xxx.adv.br**,**zzz.eng.br** e **xxx.zzz.nom.br**, mas não podem registrar **xxx.adv.br** e **xxx.eng.br**. A seguir podemos ver o Quadro 1.3 com algumas definições de nomes de domínios internet.

Aprenda como registrar um domínio para a internet acessando http://www.videolog.tv/video.php?id=376141

Assista ao vídeo disponível em http://www.youtube.com/watch?v=D0bZDGW-KQo e poste no AVEA um texto sobre o porquê da necessidade de registrar um domínio na internet.

Quadro 1.3: Definições de domínios na internet

mil	br	Brasil	**Militar**
edu	au	Austrália	**Universidade**
com	ca	Canadá	**Comercial**
gov	de	Alemanha	**Governo**
net	uk	Inglaterra	***Gateway/Host***
org			**Outras Organizações**

Fonte: Elaborada pelo autor

1.7.9 URL

De acordo com Kuwabara (2012) todos os recursos disponíveis na **www** têm um endereço único. Esse endereço é sua URL (*Uniform Resource Locator*). Através de URLs torna-se possível acessar *home pages*, arquivos disponíveis para FTP, aplicações que permitem a composição de mensagens de correio eletrônico, computadores remotos (Telnet), sistemas de menu Gopher, bancos de dados Wais, arquivos locais, etc. O endereço da URL é assim interpretado:

Seja o endereço:

http://Ibase.org.br:80/campanhas/cidadania/fome.htm#LOCAL

|-1---|----2---------|-3-|4|---------5-------------------|-------6-----|----7----|

Ele aponta para um local específico dentro de uma página escrita em HTML e é composto por seis campos. No entanto, nem todas as URLs necessitam ser tão completas. Muitas vezes bastam dois ou três desses campos para indicar "aonde" e "como" se quer chegar. Vamos analisar cada parte dessa URL. Protocolo é o tipo de serviço que queremos acessar na www. Os protocolos são:

Aprenda a configurar a rede no Windows 7, acessando http://www.todoespacoonline.com/montando-e-configurando-uma-rede-no-windows-7___138

Acesse um *site* interessante sobre vários assuntos de redes de computadores http://www.redescomputadores.com/como-montar-um-computador-dicas-de-manutencao-de-pc/

Aprenda a história da internet acessando http://www.youtube.com/watch?v=B0VY3jI1D9Y

Aprenda a evolução da internet acessando http://www.youtube.com/watch?v=-G3VZzQKNhE

Assista à vídeo aula de como montar uma rede ponto a ponto em http://www.mxmasters.com.br/video-aulas/redes/curso-de-redes-de-computadores-introducao e http://www.mxmasters.com.br/video-aulas/redes/redes-mini-curso-criando-uma-rede-ponto-a-ponto-gratis/

Conheça a nova internet que está surgindo, a WEB 2.0, acessando: http://www.youtube.com/watch?v=NJsacDCsiPg&feature=related

- http:// – para acessar uma página.
- ftp:// – para *File Transfer Protocol*.
- gopher:// – para Gopher.
- news:// – para acessar um grupo da Usenet através do protocolo NNTP.
- telnet:// – para conectar a um computador remoto.
- wais:// – para bancos de dados indexados.
- file:// – para arquivos locais.

1. **Nome do domínio ou *site***: é o nome do domínio onde o recurso está localizado. Muitas vezes o nome de um domínio nos fornece informações interessantes. Sua sintaxe de forma geral é:

   - UmOuMaisNomesSeparadosPorPontos.TipoDoDomínio.País.
   - Domínio é relacionado com o tipo de localidade onde está instalado o serviço.

2. **País:** é a sigla de países, a qual é composta de duas letras (Veja exemplo no Quadro 1.3).

   - Páginas com a terminação.br estão localizas em território brasileiro.
   - Páginas que não possuem terminação indicando o país de origem estão situadas nos Estados Unidos.
   - No Brasil, quando o tipo do domínio não é citado, a instituição é acadêmica, como cefetmg.br.

Assista ao vídeo em http://olhardigital.uol.com.br/produtos/central_de_videos/como-a-internet-funciona . Poste no AVEA, na forma de fórum, suas impressões sobre como funciona a internet.

3. **Porta:** é usado para identificar o serviço que está sendo executado.

   - Os valores da tabela da Figura 1.17 são considerados padrões e, portanto, não precisam ser colocados no endereço da URL.

4. **Diretório:** especifica em que diretório o recurso está situado.

5. **Nome:** é o nome do recurso requerido. Normalmente, páginas de www têm terminações .html ou .htm.

6. **Local:** uma página pode ser bastante longa. Muitas vezes, é interessante remeter ao usuário uma parte específica do documento. O "local" indica qual é a parte dentro da página que deve ser exibida.

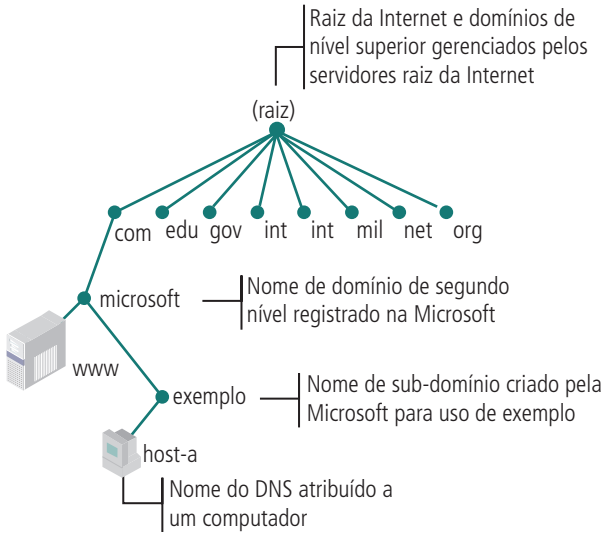


Figura 1.17: Exemplo de hierarquia de domínios
Fonte: http://technet.microsoft.com/pt-br/library/cc737203%28WS.10%29.aspx

Quadro 1.4: Portas mais comuns do TCP/IP

Serviço	Porta	Protocolo
FTP	21	TCP
TELNET	23	TCP
E-MAIL	25	TCP
DNS	53	TCP
GOPHER	70	TCP
HTTP (WWW)	80	TCP
NEWS	119	TCP

Fonte: Elaborado pelo autor

Resumo

Nesta aula você viu os conceitos sobre o modelo RM-OSI. Conheceu os protocolos e serviços que compõem a suíte TCP/IP. Depois, pôde comparar o modelo RM-OSI com a suíte TCP/IP e viu que, apesar de ter menos camadas que o RM-OSI, a suíte se enquadra no perfil de funcionamento dele. Depois estudou o protocolo Ethernet e viu que ela atua na camada 2 do modelo RM-OSI e que possui três subcamadas. Então, estudou o surgimento da internet e conheceu os órgãos reguladores dela. Estudou ainda as suas definições, divisões de seus serviços e suas classificações. Por último viu a necessidade de registrar os nomes internet e como isso é feito.

Atividades de aprendizagem

1. Pesquise no material estudado e responda às questões a seguir. Em seguida poste suas respostas no AVEA da disciplina.

a) Pesquise os padrões 100BASE-TX, 1000BASE-T, 1000BASE-SX, 1000BASE-LX e 1000BASE-CX.

b) Qual o padrão Ethernet mais utilizado hoje?

c) Explique o funcionamento detalhado do CSMA/CD.

d) É possível alterar o MAC Address?

e) Relacione os cabos UTP e os padrões Ethernet.

f) Pesquise:

- O modo de comunicação promíscua.
- 100BASE-TX.
- As diferenças entre 1000BASE-T e 1000BASE-SX.

g) Quais as semelhanças entre o modelo OSI e a Arquitetura TCP/IP? E as diferenças?

h) O que é endereço MAC?

i) Qual a relação entre um endereço MAC e um fabricante de placas de redes computacionais?

j) O que é LLC? Quais são os campos que o LLC possui?

k) Qual a finalidade de definir domínios na internet?

l) Pesquise: qual é o órgão no Brasil responsável por gerenciar nomes internet?

m) Pesquise: quais os passos que devem ser seguidos para registrar um domínio na internet no Brasil?

n) O que é uma URL e quais campos a compõem?

Aula 2 - O protocolo TCP/IP

Objetivos

Conhecer os protocolos que compõem a suíte TCP/IP.

Conhecer os serviços *web*.

Como definimos na aula anterior, a suíte TCP/IP é um conjunto de protocolos que controla a comunicação de dados em redes de computadores. Ele é dividido em quatro camadas. Nesta aula abordaremos os principais protocolos que compõem o conjunto TCP/IP de protocolos. Vale lembrar que alguns desses protocolos são confundidos pela própria aplicação que os utiliza.

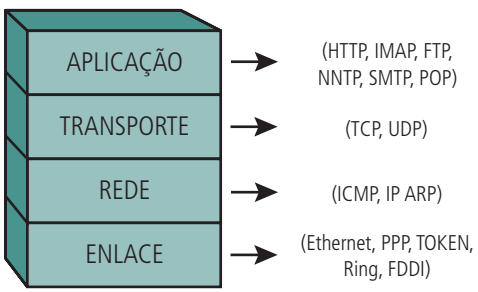


Figura 2.1: Camadas do Modelo TCP/IP e seus respectivos protocolos
Fonte: Coutinho (2010)

Como já vimos, as quatro camadas da suíte TCP/IP são:

- Camada de aplicações: protocolos de aplicações: FTP, Telnet, DNS, SNMP, SMTP, entre outros.
- Camada de transporte: protocolos de transporte: TCP, UDP.
- Camada de rede ou internet: protocolos de endereçamento de rede: IP, ARP, RARP, ICMP.
- Camada de enlace: (acesso à rede, interface de rede ou *fata-link*): protocolos de acesso ou enlace físico: CSMA/CD-Ethernet, PPP, HDCL, Token-ring, FDDI,

que interagem com o *hardware* e o meio de transmissão, permitindo que as camadas de cima independam do meio de transmissão utilizado.

- Camada física: no modelo TCP/IP, que é composta pelo *hardware*, sinais elétricos, meios de transmissão e seus padrões.

2.1 Protocolos da camada física da suíte TCP/IP

2.1.1 ARP (*Address Resolution Protocol*)

0 ARP (*Address Resolution Protocol*) permite que um *host*, ou computador, encontre o endereço físico de um host de destino na mesma rede física, apresentando somente o endereço IP de destino (COMER, 2007). Então podemos dizer que o protocolo ARP é utilizado para o mapeamento dinâmico do endereço IP, construindo uma tabela que determina o endereço da camada de enlace, ou endereço físico (MAC) correspondente ao endereço IP. Pois em redes locais a comunicação não se dá pelo protocolo IP, e sim pelo endereço físico (MAC).

Para determinar um endereço de destino de um datagrama IP, é necessário consultar a tabela ARP para obter o endereço MAC correspondente a ele. Se o endereço não estiver presente na tabela, o protocolo ARP envia um *broadcast* para todas as estações da rede, procurando a estação de destino. O destinatário que tiver o endereço IP informado responde (à máquina solicitante) seu endereço físico. Nessa ocasião, tanto a tabela da máquina origem, quanto à da máquina destinatária são atualizadas com os endereços.

2.1.2 RARP (*Reverse Address Resolution Protocol*)

Segundo Soares (1995), de forma inversa ao protocolo ARP, uma máquina utiliza o protocolo RARP para procurar um endereço IP relacionado a um endereço físico (MAC) determinado.

Segundo Comer (2007), a maior finalidade do protocolo RARP é fornecer Endereço IP para uma máquina através de seu endereço MAC. Assim, é necessário, para que o RARP funcione, que exista na rede local ao menos um servidor RARP.

Da mesma forma que o ARP, o RARP envia uma mensagem *broadcast* solicitando o endereço IP. Caso haja mais de um servidor RARP, um deles é determinado como prioritário, onde será feita a primeira pesquisa. Se dentro de um intervalo de tempo não houver respostas, outros servidores iniciarão a pesquisa.

O DHCP é uma implementação moderna do RARP. Falaremos do DHCP com mais detalhes adiante.

2.2 Protocolos da camada de rede da suíte TCP/IP

Segundo Torres (2009), a camada de internet do modelo TCP/IP é equivalente à camada 3 do modelo RM-OSI. Assim, todas as explicações dadas sobre essa camada são válidas aqui. Às vezes essa camada é também chamada de camada de *internetwork*.

Ainda de acordo com Torres (2009), há vários protocolos que podem operar nessa camada: IP (*Internet Protocol*), ICMP (*Internet Control Message Protocol*), DHCP (*Dynamic Host Configuration Protocol*), BOOTP (Bootstrap Protocol) entre outros. A seguir serão descritos alguns desses protocolos.

2.2.1 IP (*Internet Protocol*)

O IP é um protocolo usado entre duas ou mais máquinas em rede para encaminhamento dos dados. Os dados numa rede IP são enviados em blocos referidos como pacotes ou datagramas.

O IP fornece um serviço de datagramas não confiável, ou seja, o pacote vem quase sem garantias. O pacote pode chegar desordenado, duplicado, ou pode ser perdido por inteiro. Se a aplicação precisa de confiabilidade, esta é acionada na camada de transporte.

Características básicas do protocolo IP, segundo Kuwabara (2012), são:

- A entrega do pacote IP não é garantida pelo protocolo, ou seja, podem ocorrer perdas de pacote ao longo da transmissão.
- É um protocolo não orientado à conexão.
- Os pacotes podem se perder, atrasar ou ser entregues fora de ordem.

- Cada pacote pode seguir diferentes rotas (caminhos).

- Os pacotes IP têm um tempo de vida para trafegar na rede, após o qual, não alcançando seu destino, são descartados pela rede para não ficar vagando e ocupando seus recursos.

2.2.2 ICMP (*Internet Control Message Protocol*)

O protocolo ICMP é utilizado para transmissão de mensagens de controle ou de ocorrência de problemas. Utiliza o protocolo IP para o transporte das mensagens. Geralmente as mensagens ICMP são geradas pelos *gateways*, podendo também ser geradas pela estação destinatária. No caso de problemas com datagramas enviados pela estação de origem, o ICMP inclui no seu datagrama de ocorrências o cabeçalho além de 64 *bits* iniciais dos dados do datagrama IP que originou o erro.

O uso mais comum do protocolo ICMP é pelos utilitários ping, que envia um datagrama ICMP a um host na rede para verificar se ele está disponível e o tracert que envia datagramas ICMP para cada nó na rede, no caminho até chegar ao host destinatário, para determinar a rota que deve ser seguida para alcançar esse host. Um exemplo do utilitário tracert pode ser visto na Figura 2.2.

As ocorrências do ICMP, segundo Comer (2007), podem ser:

- Destinatário inacessível.

- Ajuste de fonte – solicita à estação a redução da taxa de emissão de datagramas.

- Redireção – é a rota mais adequada para a estação destinatária (para atualização da tabela de endereço dos roteadores).

- Eco e resposta de eco.

- Tempo excedido.

- Problemas de parâmetros.

- Marca de tempo e resposta à marca de tempo.

- Solicitação de informações e respostas de informações.

- Solicitação de máscara de endereço e resposta à máscara de endereço.

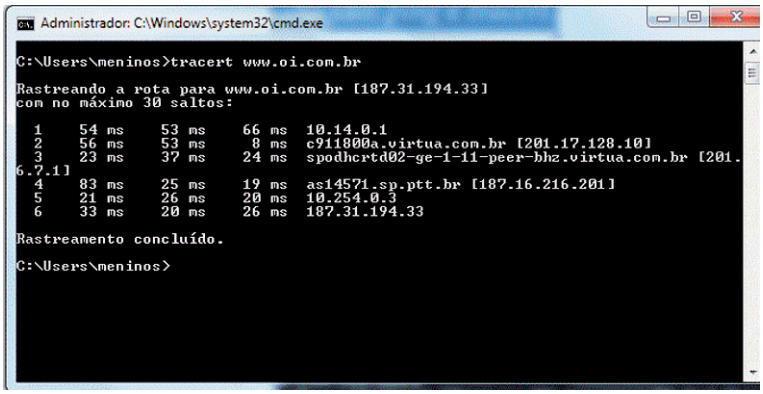


Figura 2.2: Comando tracert
Fonte: Elaborada pelo autor

2.2.3 DHCP (*Dynamic Host Configuration Protocol*)

O protocolo DHCP é responsável pelo controle e pela disponibilização de endereços IPs para os clientes. O uso do protocolo é muito difundido quando não se quer configurar um endereço IP fixo para cada computador dentro da rede – o trabalho é feito automaticamente pelo protocolo. Geralmente, o protocolo DHCP roda no servidor da rede.

2.2.4 O BOOTP (*Bootstrap Protocol*)

O BOOTP é um protocolo que surgiu para suprir algumas deficiências do protocolo ARP e permite a configuração automática de parâmetros de rede, porém sem a capacidade de alocar dinamicamente esses parâmetros, como faz o DHCP. Utiliza uma comunicação não confiável para obter seus dados

2.3 Protocolos da camada de transporte da suíte TCP/IP

Segundo Soares (1995), a camada de transporte é responsável pela movimentação de dados, de maneira eficiente e confiável entre processos (usuários). Os protocolos da camada de transporte na suíte TCP/IP são o TCP e o UDP. A seguir falaremos sobre as principais características desses dois protocolos.

2.3.1 TCP (*Tramissionr Control Protocol*)

De acordo com Comer (2007), o protocolo TCP fornece um serviço *full-duplex* e é confiável, pois provê um serviço de transporte fim a fim, ou seja, entre a origem e o destino final da mensagem, constituindo uma conexão lógica e confiável entre eles, com controle de fluxo e controle da sequência de pa-

cotes enviados e recebidos de forma a detectar eventuais perdas de pacotes ao longo da transmissão. Faz também os avisos de recebimento de pacotes.

As características fundamentais do TCP, de acordo com Lages (2012), são:

- **Orientado à conexão**: a aplicação envia um pedido de conexão para o destino e usa a conexão estabelecida para transferir dados.

- **Ponto a ponto**: uma conexão TCP é estabelecida entre dois pontos.

- **Confiabilidade**: o TCP usa várias técnicas para proporcionar uma entrega confiável dos pacotes de dados, que é a grande vantagem que tem em relação ao UDP, e motivo de seu uso extensivo nas redes de computadores. Ele permite a recuperação de pacotes perdidos, a eliminação de pacotes duplicados, a recuperação de dados corrompidos, e pode recuperar a ligação em caso de problemas no sistema e na rede.

- ***Full Duplex***: é possível a transmissão simultânea em ambas as direções (cliente-servidor) durante toda a sessão.

- ***Handshake:*** mecanismo de estabelecimento e finalização de conexão, permitindo a autenticação e encerramento de uma sessão completa. O TCP garante que, no final da conexão, todos os pacotes foram bem recebidos.

- **Entrega ordenada**: a aplicação faz a entrega ao TCP de blocos de dados com um tamanho arbitrário num fluxo de dados, tipicamente em octetos. O TCP parte esses dados em segmentos de tamanhos especificados pelo valor MTU (*Maximum Transmission Unit*). Porém, a circulação dos pacotes ao longo da rede (utilizando um protocolo de encaminhamento, na camada inferior, como o IP) pode fazer com que os pacotes não cheguem ordenados. O TCP garante a reconstrução do fluxo no destinatário mediante os números de sequência.

- **Controle de fluxo**: o TCP usa o campo janela ou *window* para controlar o fluxo. O receptor, à medida que recebe os dados, envia mensagens ACK (*Acknowledgement*), confirmando a recepção de um segmento; como funcionalidade extra, essas mensagens podem especificar o tamanho máximo do *buffer* no campo janela do segmento TCP, determinando

a quantidade máxima de *bytes* aceita pelo receptor. O transmissor pode transmitir segmentos com um número de *bytes* que deverá estar confinado ao tamanho de janela permitido: o menor valor entre sua capacidade de envio e a capacidade informada pelo receptor.

Como o TCP é um protocolo orientado à conexão, ele requer um estabelecimento da conexão entre a origem e o destino antes do início da transferência de dados.

O protocolo TCP utiliza canais lógicos, também chamados de *ports* ou *sockets*, para passar os dados das diferentes aplicações transportadas. O número da *port* identifica a aplicação para a qual o TCP está transportando os dados.

Várias aplicações podem operar simultaneamente numa conexão TCP/IP em um computador, e cada uma delas tem o seu canal lógico ou *port* específico, dentro da mesma conexão física.

Ao todo são 65.535 (64k) portas, sendo que de 0 a 1024 são portas definidas e, portanto, só podem ser usadas por aplicações que utilizem os respectivos protocolos. As portas de 1024 a 65535 são atribuídas dinamicamente.

2.3.2 UDP (*User Datagram Protocol*)

Existem situações em que o dispositivo origem não precisa da garantia de chegada dos dados no dispositivo destino; como exemplo, podemos citar alguns tipos de videoconferência. Nesses casos, o TCP é substituído pelo UDP que, segundo Lages (2012), é um protocolo que não é orientado à conexão, ou seja, não necessita estabelecer uma conexão entre origem e destino antes de enviar os dados. Ele não verifica nem se o dispositivo destino está *on-line*.

Na realidade, o protocolo UDP empacota os dados e os envia para camada inferior (rede) para que o protocolo IP dê prosseguimento ao envio. Esses pacotes, segmentos, apesar de serem numerados antes de serem enviados, não sofrem nenhuma verificação de chegada ao destino.

Assim como fizemos um paralelo entre TCP e o telefone, podemos comparar o UDP com o correio regular. Preparamos uma carta, envelopamos, selamos e colocamos no correio na esperança de que chegue ao seu destino.

2.4 Protocolos da camada de aplicação da suíte TCP/IP

De acordo com Soares (1995), na suíte TCP/IP as aplicações são implementadas de forma isolada, não existindo um padrão que defina como deve ser estruturada, baseada na arquitetura cliente/servidor.

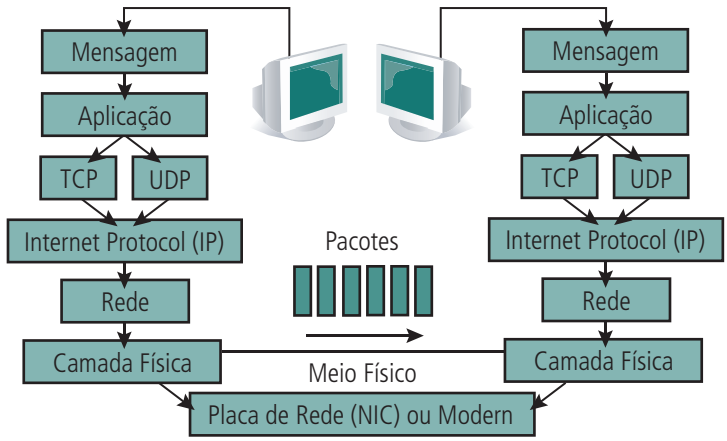


Figura 2.3: Estrutura básica do TCP/IP
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-redes-ip/RedesIP-Cap-01.pdf

A seguir faremos uma apresentação resumida das principais aplicações da suíte TCP/IP.

2.4.1 HTTP (*Hypertext Transfer Protocol*)

O *Hypertext Transfer Protocol* (HTTP), ou Protocolo de Transferência de Hipertexto, é um protocolo de aplicação utilizado para sistemas de informação de hipermídia distribuídos e colaborativos. Ele é o protocolo responsável pelo acesso às paginas *web* ou www (*world wide web*). Seu uso tem a facilidade de incorporar várias ferramentas gráficas que tem a finalidade de tornar as paginas web mais dinâmicas e atraentes para o usuário final.

Como todo aplicativo para Internet, ele funciona baseado na estrutura cliente (requisição) e servidor (resposta). O *browser*, que é o programa cliente, estabelece uma conexão com o servidor, que possui a aplicação HTML, e envia-lhe uma requisição, via URL. O servidor reponde enviando a pagina *web*, contendo o código HTML. Ao chegar no cliente, o browser requisitante monta a pagina *web* gráfica que é então visualizada pelo usuário final.

O HTTP é o protocolo que faz a comunicação entre o *browser* (programa cliente responsável pelo recebimento de páginas *web*) do computador e o servidor *web*.

Protolo TCP/IP

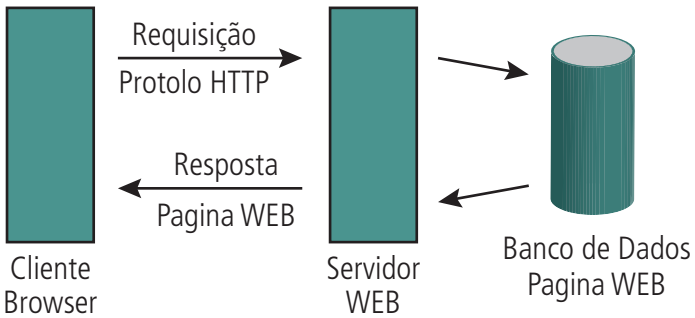


Figura 2.4: Comunicação entre o *browser* e o servidor HTTP
Fonte: http://pt.kioskea.net/contents/internet/http.php3

O HTML é um protocolo em que as informações são enviadas e recebidas conforme sãs escritas. Para o envio e recebimento de informações criptografadas, é comum a utilização de uma variação do protocolo HTTP, que é o HTTPS (*Hypertext Markup Language Secure*), o qual funciona numa porta diferente (443 em vez da porta 80).

2.4.2 DNS (*Domain Name System*)

Como foi visto no capitulo anterior, cada *home page* ou cada serviço acessado na internet e cada computador ligado à internet têm um endereço IP que os identifica. Como veremos na Aula 5, o endereço IP é um numero com quatro octetos do tipo XXX.XXX.XXX.XXX e, consequentemente, de difícil memorização. A função do serviço de DNS é converter esses endereços IP em nomes simbólicos, que são mais fáceis de manipular ou memorizar. Fazendo isso, os equipamentos na internet podem ser referenciados por um nome simbólico associado ao seu endereço IP. É o protocolo DNS que gerencia a hierarquia de domínios URL (*Uniform Resource Locator*).

2.4.3 Correio eletrônico

De acordo com Cyclades (2000), o correio eletrônico é o serviço que permite a troca de mensagens entre usuários através da internet. Permite também a troca de mensagens com usuários de outras redes de serviços (CompuServe, America Online, BITNET, FidoNEt) e com usuários de redes corporativas, não totalmente interligados à internet. Uma grande vantagem no seu uso é que ele permite anexar à sua mensagem vários tipos de arquivos como fotos, vídeos, som, ou arquivos executáveis.

O funcionamento do correio eletrônico tem como base um endereço conhecido como *e-mail address* ou endereço de correio eletrônico, cujo formato é: user@host (CYCLADES, 2000),no qual:

- *user* - representa o identificador de uma caixa postal (um espaço em disco – usuário) para recebimento de mensagens;

- *host* - representa o nome do domínio do equipamento que pode localizar essa caixa postal; esse endereço pode estar associado a um usuário, a um grupo de usuários ou mesmo a um serviço a ser prestado usando o correio eletrônico como meio de transporte.

O funcionamento do correio eletrônico é baseado no paradigma *store-and-forward*, onde os usuários envolvidos na transferência de uma mensagem não interagem diretamente entre si, e sim com programas servidores encarregados de executar e gerenciar essa transferência.

Para enviar ou receber uma mensagem de correio eletrônico, é necessário o uso do cliente de correio, o qual é chamado de agente de uso de correio (MUA ou *Mail User Agent*). Quando uma mensagem é enviada, o MTS (*Message Transfer System* – Sistema de Transferência de Mensagens) de um servidor usa o agente de transferência de mensagens (MTA ou *Mail Transfer Agent*) para examinar o endereço da pessoa para a qual a mensagem está sendo enviada. Se a pessoa for achada na rede do remetente, a mensagem é entregue a um agente de entrega de mensagens (MDA ou Agente de Entrega de Mensagens). O MDA então entrega a mensagem à pessoa certa. Quando você envia uma mensagem para alguém em outra rede na internet, a mensagem é enviada pelo MTA através da internet. A mensagem frequentemente tem que viajar através de uma série de redes antes de chegar ao seu destino, redes essas que podem usar formatos diferentes de correio eletrônico. Portas de comunicação não estão vinculadas a uma máquina em particular ou a uma combinação de *hardware* e *software*, nem restritas ao processamento de correio eletrônico. Elas podem executar uma variedade de tarefas além da conversão de protocolos. Um exemplo é a tradução de dados de um formato para outro, como eles fazem na conexão de computadores pessoais com computadores de grande porte.

Há alguns protocolos padrão que são utilizados para o envio e recebimento de correspondência eletrônica (*e-mail*). Dentre esses protocolos, destacamos:

- SMTP (*Simple Mail Transfer Protocol*): é um padrão internacional utilizado para transferência de correspondências eletrônicas (*e-mails*) entre computadores.

- POP (*Post Office Protocol*): é um dos protocolos utilizados por leitores de *e-mail* (Eudora, Netscape, Outlook, etc.) para buscar mensagens no servidor de *e-mail*. As mensagens são transferidas do servidor para o computador local quando o usuário se conecta ao servidor.

- IMAP (*Internet Message Access Protocol*): é outro protocolo padrão utilizado por leitores de *e-mail* para ter acesso às mensagens que chegam ao servidor de *e-mail*. Diferentemente do POP, o IMAP não envolve a transferência das mensagens para o computador cliente.

- Pode-se usar os seguintes canais para os serviços de correio eletrônico:

- Serviço compartilhado para usuários: usado para o envio e recebimento de mensagens via programas de *e-mail*, como: Outlook Express, Outlook, Thunderbird e o *webmail* do domínio.

- Serviço compartilhado para *websites*: usado para envio e recebimento de *e-mails* disparados por *websites* ou páginas de web como, por exemplo: www.gmail.com.

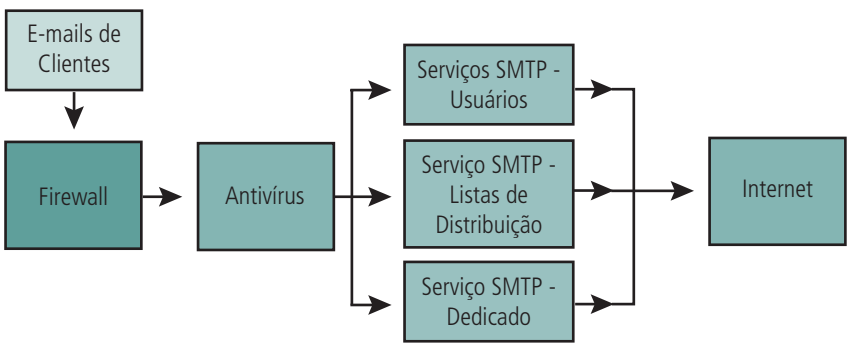


Figura 2.5: Serviço de envio via cliente de correio eletrônico
Fonte: http://centralserver.com.br/wiki/index.php/Qual_%C3%A9_a_estrutura_do_servi%C3%A7o_de_Correio_Eletr%C3%B4nico_para_envio_de_e-mails%3F

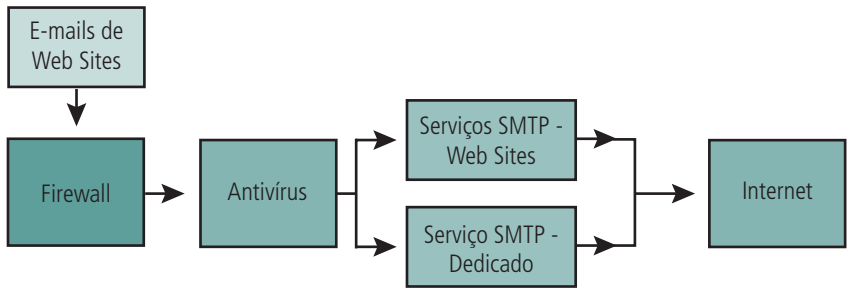


Figura 2.6: Serviço de envio via cliente de *webmail*
Fonte: http://centralserver.com.br/wiki/index.php/Qual_%C3%A9_a_estrutura_do_servi%C3%A7o_de_Correio_Eletr%C3%B4nico_para_envio_de_e-mails%3F

2.4.4 Telnet

Os protocolos de fluxo confiável, como o TCP/IP, tornam possível a conexão remota entre dois computadores, emulando um terminal dedicado. Este serviço é chamado de Telnet.

O Telnet permite a um usuário em uma máquina estabelecer uma conexão TCP com um servidor e enviar uma sequência de caracteres do teclado do usuário diretamente para a máquina servidora, como se não houvesse distância entre elas. O protocolo também carrega de volta os resultados das ações na forma de caracteres na tela do terminal. O serviço é considerado transparente, pois dá ao usuário a impressão de que sua máquina é que está executando os processos solicitados.

O protocolo oferece três serviços básicos. Primeiramente, ele age como um terminal virtual de rede, que oferece uma interface compatível com a do sistema remoto. A aplicação cliente não precisa entender os detalhes do sistema remoto, ela precisa apenas emular as teclas e a saída de tela. Em segundo lugar, o Telnet inclui um mecanismo que permite aos sistemas conectados negociar um conjunto de opções (por exemplo, *bytes* de sete *bits*, paridade, *stop-bit*). Por fim, o protocolo trata as duas pontas da conexão igualmente, isto é, ele não obriga que as entradas do cliente venham através do teclado, nem exige que as respostas do servidor sejam exibidas na tela. Com isso, o Telnet permite que outras aplicações, diferentes da emulação de terminal, utilizem os seus recursos para se comunicarem.

Este protocolo vem sendo gradualmente substituído pelo SSH, cujo conteúdo é criptografado antes de ser enviado e oferece mais segurança na comunicação.

2.4.5 SSH (*Secured Shell*)

O SSH inicialmente permitia executar apenas comandos de texto remotamente; depois passou a permitir executar também aplicativos gráficos e, em seguida, ganhou também um módulo para transferência de arquivos, o SFTP. A vantagem do SSH sobre o Telnet e o FTP é que tudo é feito através de um canal encriptado, com uma excelente segurança.

O SSH pode ser usado também para encapsular outros protocolos, criando um túnel seguro para a passagem dos dados, o que torna possível acessar servidores de FTP, proxy, *e-mail*, *rsync*, etc. de forma segura. Graças a isso, o SSH é usado como meio de transporte por diversos programas. O sistema de encriptação utilizado pelo SSH, assim como os túneis encriptados, trabalham no nível 6 do modelo OSI, acima da camada de sessão, do protocolo TCP/IP, e de toda a parte física da rede.

2.4.6 FTP (*File Transfer Protocol*)

O Protocolo de Transferência de Arquivos (FTP) é um sistema que possibilita a transferência de arquivos binários entre computadores conectados a uma rede usando o protocolo TCP. A transferência de arquivos pode ocorrer em ambas as direções. Pode-se recuperar arquivos de um servidor remoto ou enviar arquivos para um repositório ao qual se tenha acesso. Repositórios FTP são discos rígidos de computadores, ou porções destes, separadas para o armazenamento de arquivos com o propósito expresso de compartilhá-los com outros. Usuários internet podem obter cópias desses arquivos presumindo que esses possuam o endereço do servidor e as senhas de acesso ao sistema ou que utilizem o sistema chamado FTP *anonymous*, que permite a qualquer usuário entrar no sistema como convidado (*guest*) e recuperar arquivos.

Arquivos executáveis (.exe .com), arquivos comprimidos ou documentos fora do padrão ASCII são todos arquivos binários e devem ser transmitidos como tal. Essas operações são normalmente conhecidas como *download* (baixar) e *upload* (enviar)

O FTP pode ser feito diretamente pelo *browser web* usando o protocolo HTTP e carregando uma página comum ou pela digitação de protocolo FTP no mesmo *browser*.

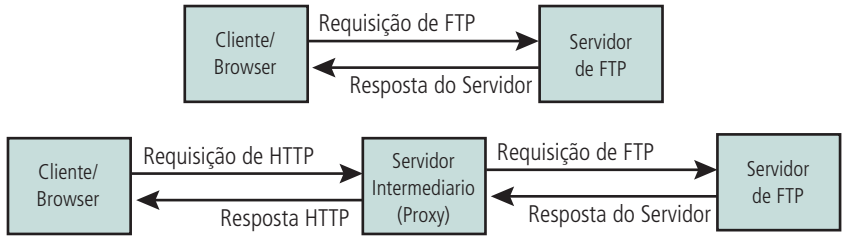


Figura 2.7: Modos de fazer FTP
Fonte: Elaborada pelo autor

2.4.7 TFTP (*Trivial File Transfer Protocol*)

O TFTP (*Trivial File Transfer Protocol* – Protocolo Trivial de Transferência de Arquivo) é um protocolo que desempenha o mesmo tipo de aplicação (transferência de arquivos), mas, ao contrário do FTP, usa o protocolo UDP (porta 69) na camada de transporte, não usando o gerenciamento da entrega dos pacotes, que ficará a cargo do usuário final.

Por não usar os mecanismos de confirmação e reordenamento de pacotes, os pacotes UDP são menores (já que o cabeçalho UDP é menor do que o cabeçalho TCP) e necessitam de menos poder computacional para serem processados, já que o processo de reordenamento e confirmação de recebimento não são necessários. Será a aplicação, e não o protocolo, que se encarregará de executar essas funções.

Para aplicações cotidianas, o protocolo TFTP não tem muita utilidade, já que o FTP é muito mais confiável. No entanto, existe um tipo de aplicação que tira proveito do TFTP: o *boot* remoto.

2.4.8 Diferença entre o FTP e TFTP

O FTP e o TFTP são protocolos que podem ser usados para transferir arquivos pela internet. As diferenças entre os dois protocolos são:

- FTP é um protocolo de transferência de arquivo completo, orientado por sessão, garantido a entrega do arquivo. TFTP é usado como um protocolo de transferência de arquivo de finalidade básica, não orientado à conexão e por isso não garante a entrega do arquivo.

- FTP pode ser utilizado interativamente. TFTP permite apenas transferência unidirecional de arquivos.

- FTP depende do TCP, é orientado por conexão e fornece controle confiável. TFTP depende UDP, requer menos sobrecarga e praticamente não fornece controle.

- FTP fornece autenticação de usuário. TFTP, não.

- FTP usa números de porta TCP conhecidos: 20 para dados e 21 para caixa de diálogo de conexão. TFTP usa o número de porta UDP 69 para sua atividade de transferência de arquivo.

2.4.9 NFS (*Network File System*)

Trata-se de um sistema de arquivos distribuídos, desenvolvido inicialmente pela Sun Microsystems, a fim de compartilhar arquivos e diretórios entre computadores conectados em rede, formando assim um diretório virtual. Ele faz transferências de arquivos, como o TFTP, utilizando o protocolo UDP. Antes da transferência o NFS, adicionalmente, faz acesso *on-line* aos arquivos da rede.

2.4.10 SNMP (*Simple Network Management Protocol*)

É um protocolo de comunicação usado para transmitir informações de ***status*** de equipamentos conectados em rede a um servidor gerenciador dessas informações. Ele utiliza o protocolo UDP para fazer gerência de equipamentos.

Os *hosts* em uma rede possuem um *software* cliente, também chamado de agente SNMP, que recolhe informações do próprio equipamento em que estão carregados e envia para um servidor por meio do protocolo SNMP.

No servidor de gerenciamento existe um *software* chamado de gerente SNMP que recebe essas informações e as armazena numa base de dados chamada de MIB (*Management Information Base*).

Assim, o funcionamento dessas aplicações está vinculado ao envio/recebimento periódico de mensagens, equipamentos/computadores respectivamente, que contêm os valores dos parâmetros para o monitoramento, análise e posterior intervenção.

2.4.11 RPC (*Remote Procedure Call*)

Implementa mecanismos de procedimentos de chamada remota, úteis no desenvolvimento de aplicações cliente-servidor com um nível maior de abstração. Uma aplicação utiliza o RPC para fazer interface das suas funções. Assim as funções chamadas pelas aplicações são repassadas ao RPC que

monta uma mensagem correspondente e envia para processamento remoto. O servidor, então processa as mensagens, executa a rotina e devolve os resultados para o RPC da estação, que reestrutura os dados e repassa à aplicação. Tudo isso em uma função virtualmente local, transparente para a aplicação.

2.4.12 WAIS (*Wide Area Information Server*)

Os servidores de informações de área ampla (WAIS) são sistemas de recuperação de informações distribuídas. Eles auxiliam os usuários a pesquisarem bancos de dados através da internet, utilizando uma interface de fácil utilização. Os bancos de dados (conhecidos como fontes) são em sua maioria documentos de texto puro, mas podem conter também figuras ou vídeos. Os repositórios de dados podem estar organizados de diferentes formas, utilizando diversos sistemas gerenciados, porém ao usuário, não é requerido nenhum conhecimento prévio de linguagens de consulta. O WAIS utiliza-se de consultas em linguagem natural para encontrar documentos relevantes. O resultado da consulta é um conjunto de documentos que contém as palavras-chave utilizadas na consulta, sem qualquer tradução de seu significado.

Assim, como ocorre com o sistema Gopher e Veronica, o avanço do protocolo HTTP praticamente extinguiu servidores WAIS na internet, pois *sites* de buscas, como o Google, tornaram mecanismos de busca muito eficientes.

2.4.13 Network news

É um serviço composto por informações agrupadas por categorias, para maior facilidade de divulgação e acesso. É originado a partir da rede Usenet (rede acadêmica Unix, conectada através de linha discada via UCCP), amplamente difundida pela internet.

As categorias em que as informações são agrupadas são denominadas *newsgroups*, organizadas de forma hierárquica, partindo de um tipo de atividade até o assunto propriamente dito; como por exemplo, "rec.music.classic" refere-se à musica clássica. Esses grupos podem ser livres, quando não há controle sobre as informações envolvidas, ou moderados, quando há triagem dessas informações antes da publicação.

2.5 LINGUAGENS PROGRAMAÇÃO PARA *WEB*

As tecnologias associadas às aplicações *web* têm a finalidade de criar um meio onde seja possível a realização da interação entre o usuário com as

informações contidas na *web*. Baseada nesse paradigma, a *web* está construída em sistemas de *sites* (ou páginas) denominados hipertexto.

De acordo com Simon (2012), o hipertexto é um dos paradigmas básicos em que a teia mundial se baseia. Ele é uma espécie de texto multidimensional em que numa página trechos de texto se intercalam com referências a outras páginas. Clicando com o *mouse* numa referência dessas, a página corrente é substituída pela página referenciada. Os hipertextos, além de possibilitar facilidade de uso pelo usuário final, também permitem a interação de páginas com imagens, vídeos, áudio e todo tipo de apresentações, o que os leva a serem também chamados de hipermídia.

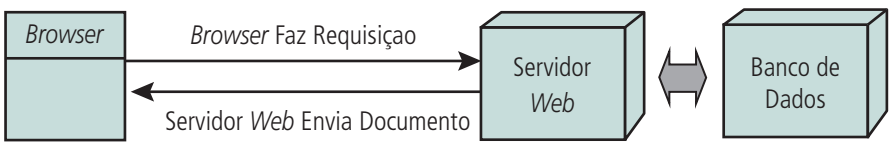


Figura 2.8: Sistema *web* simples
Fonte: https://www.jhcruvinel.com/index.php?option=com_docman&task=doc_download&gid=59&Itemid=12

Normalmente, todos os *sites* e páginas da *web* utilizam as mesmas linguagens e tecnologias básicas, seja para a exibição de documentos ou execução de eventos e funções. Para que um *site* da *web* seja visualizado numa máquina, é necessária a existência de um *browser*, ou navegador *web*, que possuam configurações padrões que possam ser alterados de acordo com a necessidade do usuário. A seguir serão apresentadas as linguagens mais usadas para programação *web*.

2.5.1 HTML (*Hypertext Markup Language*)

Existem várias linguagens específicas para o desenvolvimento de páginas web. A mais conhecida delas é a HMTL, cuja edição é simples e rápida, o que facilita a sua implementação e manutenção. Nesse padrão é possível, além de hipertextos, implementar figuras animadas e formulários a serem preenchidos.

A HTML consiste numa linguagem de marcação que permite a visualização de documentos num *browse*r através de uma *web page*. É baseada numa linguagem de *tags* definida pela SGML (*Standard Generalized Markup Language*), e define como um documento deve ser exibido na tela de um computador.

Complementando o funcionamento da HTML, são usadas outras linguagens auxiliares que permitem o tratamento e processamento de eventos no cliente. Esses eventos consistem em funções existentes numa *web page* que geralmente são solicitadas pelo próprio usuário.

2.5.2 Java Script

Java Script é uma linguagem auxiliar baseada em *script* que fica embutida no código HTML, adicionando novas funcionalidades a uma *web page*. Mesmo não sendo orientada a objetos, ela permite que seus objetos genéricos possam ser instanciados e executados, auxiliando a execução de rotinas do lado do cliente, tais como validações de dados de formulários, abertura de novas telas, criação de menus dinâmicos, criação e tratamento de *cookies*, entre outras. A maioria dos *browsers* já vem com a tecnologia Java Script embutida.

2.5.3 Java

Java é uma linguagem de programação, similar à linguagem C++, orientada a objetos. A grande vantagem dessa linguagem está no uso da Java Virtual Machine (JVM), que permite que apenas uma única versão Java compilada do programa seja executada em qualquer tipo de sistema operacional, tornando-a uma linguagem que independe da plataforma utilizada. Isso significa que um programa escrito em Java pode ser executado numa grande variedade de computadores, o que é uma vantagem considerável, pois outras linguagens requerem que o programa seja compilado separadamente para cada tipo de computador, o que resulta em várias versões diferentes do código. Isso requer uma quantidade substancial de trabalho.

Java pode servir para desenvolver vários aplicativos, além de permitir o acesso a bancos de dados. Ela pode criar aplicações interativas de multimídia também.

Um uso comum de Java é a criação de teleimpressores de notícias transmitindo as notícias mais recentes, das quais as pessoas podem obter mais detalhes clicando nelas. Isso pode ser usado em intranets para apresentar informações ou notícias. Esse processo também permite às pessoas escolherem se o querem ligado ou desligado. Dependendo do *applet* (componente de um aplicativo), a quantidade de recursos necessários vai variar (assim como a memória). Basicamente, quanto maior a *applet*, mais recursos são necessários.

A linguagem Java pode também ser usada para criar programas que ajudem pessoas a navegar mais facilmente através de uma página intranet e explorar a enorme quantidade de dados trancados em bancos de dados corporativos. Assim, essa linguagem se tornou tão importante para as aplicações de intranet que as companhias de *software* e *hardware* lançaram *add-ons* Java (componentes que podem ser adicionados) e bibliotecas para permitir que desenvolvedores de Java acessem bancos de dados legados, como os *mainframes* IBM. Isso pode acelerar a mudança em direção à linguagem Java nas intranets.

2.5.4 PHP (*Personal Home Page*)

A PHP é muito parecida, em tipos de dados, sintaxe e mesmo funções, com a linguagem C e com a C++. É uma linguagem interpretada livre e modularizada, tornando-se propícia para desenvolvimentos dinâmicos e ideal para instalação e uso em servidores *web*. Diversos módulos são criados no repositório de extensões PECL (PHP *Extension Community Library*) e alguns desses módulos são introduzidos como padrão em novas versões da linguagem. Pode ser funcionar também como linguagem auxiliar, embarcada no código HTML.

Construir uma página dinâmica baseada em bases de dados é simples com PHP, pois ela provê suporte a um grande número de bases de dados. Ela tem suporte aos protocolos: IMAP, SNMP, NNTP, POP3, HTTP, LDAP, XML-RPC, SOAP, sendo possível abrir *sockets* e interagir com esses e outro protocolos. Quando se usam bibliotecas de terceiros, é possível expandir ainda mais essas funcionalidades.

Existem iniciativas para utilizar o PHP como linguagem de programação de sistemas fixos. A mais notável é a PHP-GTK. Trata-se de um conjunto do PHP com a biblioteca GTK, portada do C++, fazendo assim *softwares* interoperacionais entre Windows e Linux. Na prática, essa extensão tem sido muito pouco utilizada para projetos reais.

2.5.5 CGI (*Common Gateway Interface*)

Em geral, os *softwares* de *web* não são particularmente amigáveis aos bancos de dados. E nos primórdios da *web*, o acesso a banco de dados era, então, uma tarefa dificílima. Mas, com o aparecimento de novas tecnologias, a tarefa de acesso aos bancos de dados ficou menos complicada devido aos vários *softwares* que foram desenvolvidos para facilitar o esse acesso pelos sistemas da internet. Dentre esses *softwares*, pode-se destacar o *Interface de Gateway Comum* (CGI).

Essencialmente, o CGI é uma interface que entrega informação de um servidor para o seu programa e, do seu programa, de volta para o cliente solicitante. Não é uma linguagem de programação, pois é o programa de acesso ao banco que faz todo o processamento. O CGI permite que qualquer *query* (pesquisa em banco de dados) possa ser executada, e tenha sua resposta enviada de volta para o cliente que a requereu. Permite também que programadores de intranet escrevam programas e *scripts* que darão acesso às pessoas para usarem seus *browsers* (navegadores) para facilmente pesquisar dados nos bancos de dados preenchendo formulários em páginas e enviando os resultados de volta em HTML, para que os *browsers* possam entendê-los. Assim ouso de CGI tem crescido.

Na internet, por exemplo, ao se preencher um formulário numa página, para se registrar e receber uma mensagem com uma senha, foi usado um CGI. Nesse processo, o CGI pegou a informação preenchida no formulário, executou várias ações nele (incluindo inserir a informação num banco de dados), e automaticamente criou a senha, e enviou a mensagem com a senha.

O CGI pode ser usado conjuntamente com uma variedade de tecnologias; o seu uso mais comum é a chamado de linguagem interpretada. Uma linguagem interpretada, como a popular Perl, é frequentemente preferida porque os *scripts* escritos com ela são fáceis de manipular, modificar e manter. E ele também pode ser acessado com linguagens de computador mais sofisticadas, como as **copiladas C**, o **C++** ou o **Fortran**.

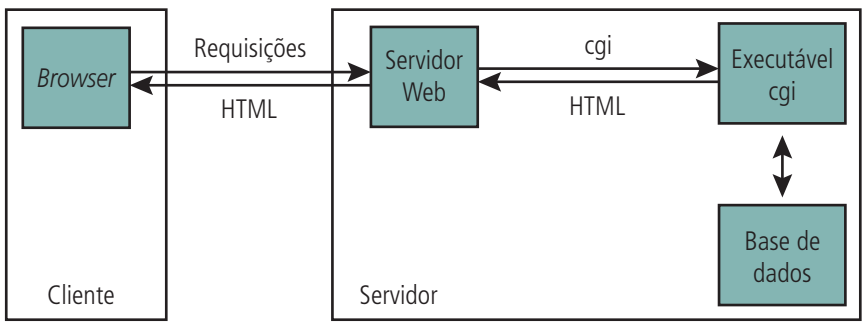


Figura 2.9: Acesso a banco de dados usando CGI
Fonte: http://www.jairo.pro.br/lpi/cgi.doc

2.5.6 Linguagem Perl

Perl é uma linguagem de programação de uso geral, desenvolvida por Larry Wall na década de 1980, baseada principalmente na linguagem C e na linguagem **awk**. Uma das principais características de Perl é facilitar a manipulação de textos e processos. Além disso, Perl possui um alto grau de portabilidade, modularidade e reusabilidade de código. Perl é uma linguagem interpretada, o que torna a carga de um programa escrito em Perl um pouco mais lenta que a carga de programas escritos em algumas linguagens compiladas. Por outro lado, isto permite uma maior flexibilidade na sintaxe da linguagem, como veremos adiante (CINTRA, 2001).

Além disso, Perl suporta orientação a objetos. E um dos seus objetivos é ajudar o programador a aumentar a produtividade escrevendo programas pequenos, mas poderosos. Existem versões gratuitas de Perl para diversas plataformas: Linux, Unix, Windows, Macintosh, OS/2, etc. Essas características fazem essa ser uma das linguagens mais usadas para a preparação de programas CGI.

2.5.7 ASP (*Active Server Pages*)

O ASP é uma estrutura de bibliotecas básicas (e não uma linguagem) para processamento de linguagens de *script* no lado servidor para geração de conteúdo dinâmico na web. O *script* é interpretado no lado do servidor e o que é enviado ao lado do usuário é apenas a saída que normalmente é uma linguagem de marcação como HTML, XHTML ou XML.

Como o ASP não é uma linguagem, é necessário o uso de linguagem em conjunto com o ele. Exemplos de linguagens aceitas pelo ASP são: VBScript, JScript, PerlScript, Python, etc., sendo apenas as duas primeiras suportadas por padrão. A linguagem ASP é nativa em servidores windows, mas pode também ser usadas em outras plataformas. Linguagens como o Javascript e o VBScript podem ser processadas pelo *browser*. Nesse caso, este tem que suportar a linguagem. Contudo, como o ASP é processado pelo servidor, há independência de *browsers*, uma vez que eles só processarão HTML.

O ASP também permite executar consultas a banco de dados, através da biblioteca de componentes ActiveX.

2.5.8 ASP.NET

ASP.NET é o sucessor da tecnologia ASP que permite, através de uma linguagem de programação integrada na .NET Framework, criar páginas dinâmicas. Também não é nenhuma linguagem de programação.

O ASP.NET é baseado no *framework* .NET, herdando todas as suas características; por isso, como qualquer aplicação .NET, as aplicações para essa plataforma podem ser escritas em várias linguagens, como C# e Visual Basic .NET.

Uma aplicação para *web* desenvolvida em ASP.NET pode reutilizar código de qualquer outro projeto escrito para a plataforma .NET, mesmo que em linguagem diferente. Ao contrário da tecnologia ASP, as aplicações ASP.NET são compiladas antes da execução, trazendo sensível ganho de desempenho.

2.6 Acesso a bancos de dados

As informações mais importantes em um sistema computacional estão normalmente guardadas em bancos de dados. Esses bancos de dados podem estar em um único local ou espalhados por todo o sistema.

A maioria dos bancos de dados existe desde muito antes da internet e isso significa que foram construídas sem se pensar no TCP/IP ou no HTML e sem levar em conta qualquer outra tecnologia internet. Porém, um sistema na internet pode, teoricamente, tornar muito mais fácil o acesso a todos os tipos de dados, pois o uso do HTML significa que é relativamente fácil construir formulários de busca que façam facilmente chegar aos dados requeridos, o que poderia ser difícil sem se saber uma linguagem de programação de bancos de dados. No entanto, enquanto é fácil construir formulários de busca em HTML que permitam às pessoas digitar pesquisas, mas não é tão fácil fazer com que essas pesquisas sejam enviadas para procurar em um banco de dados e então retornar os dados solicitados. Para fazer isso é que as ferramentas de busca *web-to-database* são projetadas para permitir que facilmente se chegue aos dados residentes em um banco de dados.

Há vários modos diferentes de acessar esses bancos de dados na internet. Um desses modos é o uso do *Common Gateway Interface* (CGI ou Interface Comum de Porta de Comunicação), descrito anteriormente.

2.7 Compartilhamento de informações

Assista ao filme Guerreiros da Net, faça uma resenha e poste no AVEA: http://pontasdamadrugada.blogspot.com/2007/03/warriors-of-net-guerreiros-da-internet.html

Uma corporação típica gera montanhas de papel todos os dias. E isso é ruim para o meio ambiente, caro para as corporações e consome muito tempo, pois requer que a companhia contrate pessoas para guardar e manter uma trilha do papel. Existem formulários que têm que ser preenchidos e manuseados, materiais de *marketing*, materiais de vendas e panfletos para serem enviados pelo correio, formulários de venda que têm que ser usados. Essa lista continua indefinidamente.

Fixe os conhecimentos adquiridos sobre o protocolo TCP/IP assistindo ao vídeo disponível em http://www.youtube.com/watch?v=XKN6CWbZk1g&feature=related

Conheça melhor os protocolos dos serviços de correio eletrônico acessando http://pt.kioskea.net/contents/internet/smtp.php3

O advento das intranets pôde diminuir em muito o uso de papel em uma corporação, fazendo surgir o "escritório sem papel". Uma combinação de tecnologias de comunicação, ferramentas de publicação na *web*, aplicações de grupo de trabalho e correio eletrônico pode ajudar nessa tarefa, além de ajudar as corporações a reagirem mais rapidamente a mudanças nos negócios e entregar bens e serviços mais rapidamente, levando a um diferencial de mercado.

Resumo

Nesta aula você conheceu os protocolos que compõem as camadas da suíte TCP/IP e como eles funcionam. Viu que eles facilitaram o uso da internet e ajudaram a torná-la o que é hoje. Baseado nesses protocolos, você viu quais são os principais serviços *web*, como são construídas as páginas *www* e como elas são vistas pelo usuário final. Aprendeu também como funciona o correio eletrônico, que é um dos grandes serviços web.

Atividades de aprendizagem

Responda às questões a seguir e poste no AVEA da disciplina:

1. Qual é o protocolo responsável pela execução remota de comandos com aplicação de requisitos de segurança?

2. Qual a relação existente entre os *browsers* e a linguagem HTML?

3. Explique como uma aplicação Telnet pode ajudar o serviço de um administrador de redes que possui vários servidores em seu domínio.

4. Quais os protocolos da camada de aplicação mais importantes para um servidor de mensagens?

5. Qual a principal diferença entre o protocolo TCP e o UDP?

6. Utilize o *prompt* de comando de seu computador para definir a diferença entre os comandos ping e tracert do Windows:

   Para executar o *prompt* de comando no Windows: Iniciar -> Executar -> cmd.exe -> OK
   Digite ping 187.31.194.33
   Digite Tracert www.oi.com.br
   Digite ipconfig e descubra o endereço Mac e ip de sua máquina.

Aula 3 – O endereçamento IPv4

Objetivos

Conhecer os endereços IPv4.

Conhecer as classes e subclasses do endereço IPv4.

Conhecer a máscara de sub-redes.

Conhecer os endereços IPv6.

Fazer uma comparação IPv4 X IPv6.

Um sistema provê um serviço de comunicação universal quando é possível a quaisquer dos elementos desse sistema se comunicar arbitrariamente. Para tornar um sistema de comunicação universal, deve-se estabelecer um método globalmente aceito para identificação dos componentes a ele conectados. Esse método deve possibilitar um sistema de identificação único, em que cada elemento ligado tem um endereço único. Assim, é possível garantir que uma mensagem chegue ao seu destino. Nesse contexto, se um *host* faz parte da internet, seu endereço deve ser único em toda a internet; assim, qualquer pacote a ele endereçado deve ser capaz de chegar ao seu destino.

Cada elemento de uma rede deve ter um único endereço para ser identificado em sua rede local e em qualquer outra rede com a qual esse elemento consegue se comunicar. Funciona assim em redes LANs e WANs e também na internet, que trabalham com a suíte TCP/IP. Na rede TCP/IP o responsável por prover esse endereço único é o protocolo IP, que trabalha na camada de rede (ou camada 3).

Esse endereço IP funciona analogicamente a um endereço físico de uma moradia. Assim como cada moradia tem o seu endereço, o mesmo acontece com os computadores que utilizam TCP/IP, em que cada um tem o seu endereço de IP único.

Segundo Torres (2009), a grande vantagem do protocolo IP é que ele é roteável, isto é, foi criado pensando-se na interligação de diversas redes, nas quais pode haver diversos caminhos interligando o transmissor e o receptor.

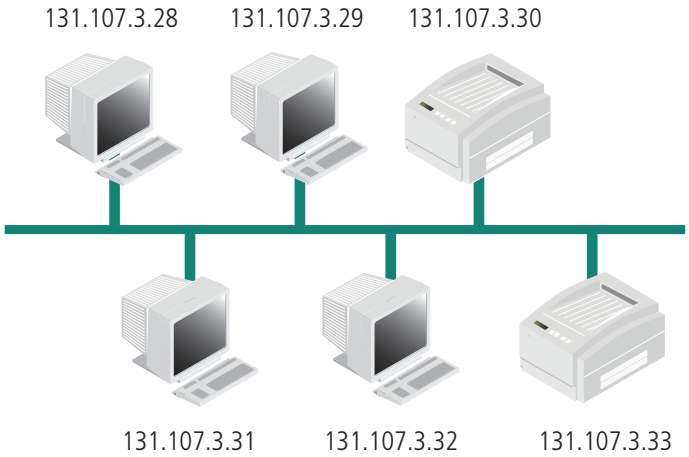


Figura 3.1: Uma rede LAN com a identificação dos elementos feita por endereços IP
Fonte: Coutinho (2010)

3.1 Definição do endereço IP

O endereço IP é um numero de 32 *bits*, representado em decimal em forma de quatro números de oito *bits* separados por um ponto, no formato "a.b.c.d". Assim, o menor endereço IP possível é 0.0.0.0 e o maior 255.255.255.255 (TORRES, 2009). Nesse endereço de 32 *bits* (ou quatro *bytes*), cada parte de oito *bits* é chamada de octeto.

Apesar de a definição dos octetos ser em *bits*, eles são normalmente representados por valores decimais, para facilitar a leitura, numa notação chamada de "notação decimal com pontos". Assim, o valor decimal de um octeto estará sempre entre 0 e 255, pois com oito *bits*, o menor valor decimal que podemos representar é o zero, e o maior, 255 (28).

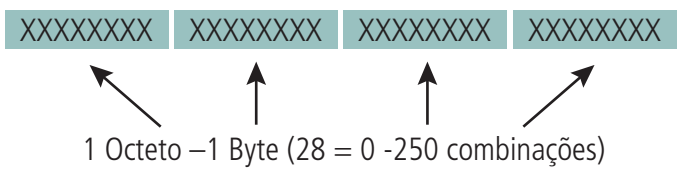


Figura 3.2: Endereço IPv4
Fonte: Elaborada pelo autor

Segundo Torres (2009), para conectar um computador à internet, é necessário obter um endereço de IP único que não esteja sendo usado por nenhum outro computador.. Esse endereço deve ser fornecido pelo provedor a qual

esse computador esteja ligado. Todavia, se o computador fizer parte de uma rede de dados doméstica privada, sem acesso à internet, o endereço IP pode ser qualquer um, lembrando apenas que cada computador dessa rede deve ter seu endereço IP único.

Tabela 3.1: Endereços de IP nas representações decimal e binária

DECIMAL	BINÁRIO
192.168.3.11	11000000.10101000.00000011.00001011
200.200.25.1	11001000.11001000.00011001.00000001
139.12.25.32	10001011.00001100.00011001.00100000
10.10.0.1	00001010.00001010.00000000.00000001

Fonte: Coutinho (2010)

De acordo com Comer (2007), o endereço IP é dividido em duas partes, que fornecem duas identificações. A primeira parte informa o endereço da rede a que o *host* pertence (*network* ID ou NetID). Lembre-se que todas as máquinas conectadas a uma mesma rede irão compartilhar esse mesmo *network* ID. A segunda parte informa o endereço do próprio *host* (host ID ou HostID). Um computador teve ter um único *host* Id e, por conseguinte, deve ter um único par NetID + HostID, ou seja um único endereço IP.

As redes computacionais normalmente são segmentadas em redes menores (LANs) para evitar excesso de tráfego e melhorar o desempenho na troca de dados entre os computadores. Essas várias LANs segmentadas e interligadas formam as WANs. Para segmentar as redes, utilizamos um dispositivo conhecido como *gateway* (roteador).

Como já foi dito, todos os *hosts* pertencentes a uma segmento (ou mesma LAN), devem possuir o mesmo NetId. E quando utilizamos roteadores para interconectar essas várias LANs em WANs, cada LAN deve possuir seu próprio NetID , que é único para cada uma delas.

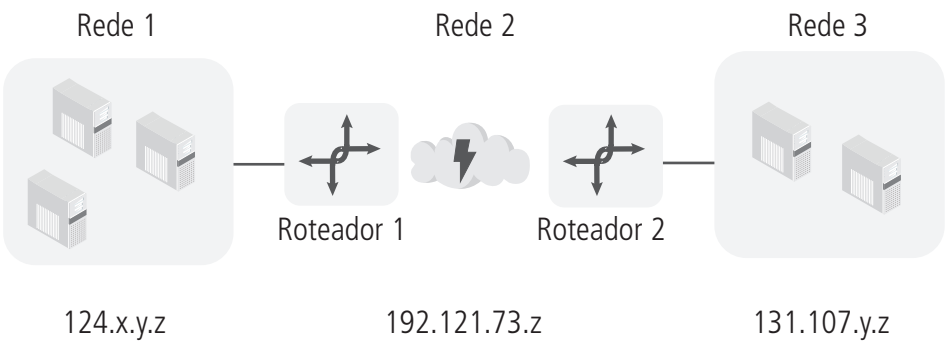


Figura 3.3: Exemplo de redes interligadas por roteadores
Fonte: Coutinho (2010)

A Figura 3.4 mostra as redes 1 e 3 roteadas, ou seja interligadas por roteadores. Perceba que a rede 1 utiliza um tipo de endereçamento de IP (124.x.y.z) e a rede 3 utiliza outro tipo de endereço (131.107.y.z), ou seja os números identificam os NetID e as letras identificam os HostID. Com isso, toda máquina ou nó da rede 1 tem o mesmo NetID (124) e todo nó da rede 3 tem o mesmo NetID (131.107), que é diferente do da rede 1.

A rede 2 representa a conexão entre as redes 1 e 3, formando um rede WAN junto com as redes LAN 1 e 3. Essa rede 2 possui um NetID exclusivo para ela, de forma que possam ser atribuídos HostIDs exclusivos para as interfaces entre os dois roteadores. A rede 2 utiliza um endereçamento IP (192.121.73) cujo NetID é diferente do das redes 1 e 3. Assim, neste exemplo temos três diferentes NetIDs para que os roteadores possam se comunicar: um para a rede 1, outro para a rede 2 e outro para a rede 3.

Conclui-se que o endereço IP identifica a rede à qual a máquina está conectada, além da máquina propriamente dita. Por isso, o IP de um computador deve informar seu HostID, para que ele possa ser identificado em sua rede local (LAN), e seu NetID, para que ele possa ser identificado em toda a rede interligada (WAN). A grande vantagem desse esquema de endereçamento IP é que ele foi cuidadosamente concebido para simplificar a tarefa de interligação entre as LANs, ou seja, o roteamento. Nessa maneira, redes que não estão fisicamente interligadas podem se comunicar (interligação virtual) através do roteamento.

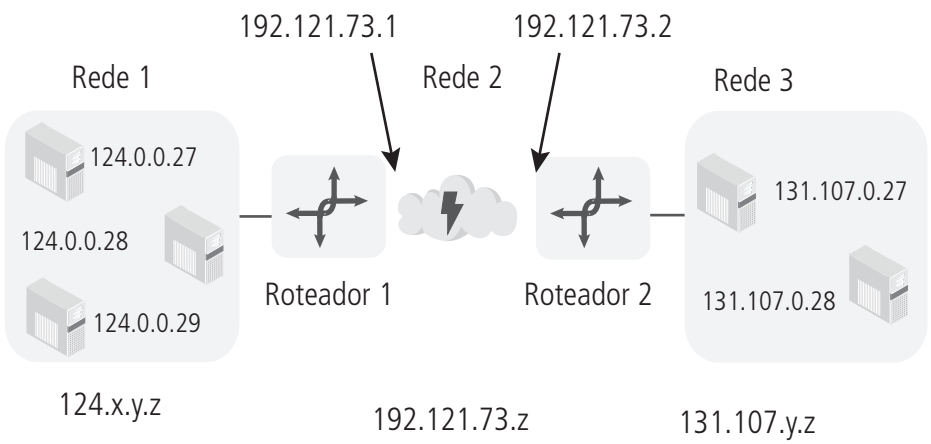


Figura 3.4: Exemplo de segmentos de redes interligadas por roteadores com endereçamento dos nós ou elementos
Fonte: Coutinho (2010)

A Figura 3.5 é baseada na Figura 3.4, com o adicionamento dos HostID em cada elemento da redes 1, 2 e 3. É mantido o roteamento entre as redes 1 e 3 através da rede 2.

A rede 1 utiliza endereçamento de NetID (124) definido como classe A; a rede 2, o endereçamento (192.121.73), definido como classe C; e a rede 3 utiliza endereçamento (131.107), definido como classe B. Essas classes serão estudadas na próxima seção.

A rede 1 possui três *hosts* e cada um deles possui o seu IP exclusivo (124.0.0.27, 124.0.0.28 e 124.0.0.29), em que o NetID é 124 para todos eles, pois fazem parte da mesma LAN, e os HostsID 0.0.27, 0.0.28 e 0.0.29. Para que esses computadores possam se comunicar com os outros segmentos de rede, terão de "falar" com o endereço 124.0.0.1 (NetID é 124 e HostID 0.0.1). Tecnicamente chamamos esse endereço de "*gateway* padrão" (*default gateway*).

A rede 2 é configurada para que os roteadores possam trocar informações entre si. O caminho entre um roteador e outro exige a configuração de sua interface. O segmento de rede de NetID 192.121.73 estabelece a conexão dos roteadores através das interfaces 192.121.73.1 (HostID 1) e 192.121.73.2 HostID 2).

A rede 3 possui dois *hosts* e cada um possui seus IPs exclusivos (131.107.0.27 e 131.107.0.28), em que o NetID é 131.107 para todos eles, pois fazem parte da mesma LAN, e os HostsID 0.27 e 0.28. Para que esses computadores possam se comunicar com outros segmentos de rede, terão de "falar" com o endereço 131.107.0.1 (NetID é 131.107 e HostID 0.1), que é o endereço do *gateway* padrão desse segmento.

LAN (*Local Area Network*): é um conjunto de computadores interligados fisicamente entre si, trocando e compartilhando informações e recursos. Tais redes são denominadas locais por cobrirem apenas uma área limitada (10 km no máximo e necessitam de roteadores).

WAN (*Wide Area Network*): é um conjunto de redes LANs, geograficamente distribuídas (que abrange uma grande área geográfica), interligadas por roteadores.

3.2 Classes de endereçamento IP

Para facilitar a distribuição dos endereços IPv4, foram especificadas cinco classes de endereço IP (TORRES, 2009), sendo três classes principais e mais duas complementares. São elas:

- Classe A:
- de 0.0.0.0 até 126.0.0.0, permitindo até 126 redes;
- cada rede permite endereçar até 16.777.214 dispositivos;
- número máximo de *hosts* em cada rede = 16.777.214 = 224 – 2;
- Classe B:
- de 128.0.0.0 até 191.255.255.255, permitindo até 16.384 redes;
- cada rede permite endereçar até 65.536 dispositivos;
- Classe C:
- de 192.0.0.0 até 223.255.255.255, permitindo até 2.097.152 redes;
- cada rede permite endereçar até 254 dispositivos;
- Classe D:
- de 224.0.0.0 até 239.255.255.255;
- o endereço de *multicast* é utilizado na transmissão simultânea de um ou mais pacotes para um grupo de *hosts*, sendo identificados por um endereço especial de destino (*multicast address*);
- Classe E:
- de 240.0.0.0 até 255.255.255.255;
- reservado.

Figura 3.5: Divisão dos endereços IP em classes
Fonte: http://www.wandreson.com/download/training-networking-tcpip.pdf

3.2.1 Redes classe A

As redes pertencentes à classe A utilizam o primeiro octeto para endereço de rede (NetID), e o *bit* mais significativo (chamados também de MSB – *Most Significant Bit*) desse octeto é sempre zero em binário. Os próximos sete *bits* é que identificam a rede (NetID). Os três octetos restantes (24 *bits*) são usados para endereçar o *host* (HostID). Essa divisão permite ter até 16.777.216 endereços de HostID.

É importante observar que o endereço de *host* não pode ser todo composto por zero ou por um, pois o endereço de *host* todo zerado é utilizado para representar o endereço de rede (NetID), e o endereço de *host* todo composto de um é utilizado para fazer *broadcasting*, que é o processo pelo qual se transmite ou difunde determinada informação para todos os *hosts* ao mesmo tempo.

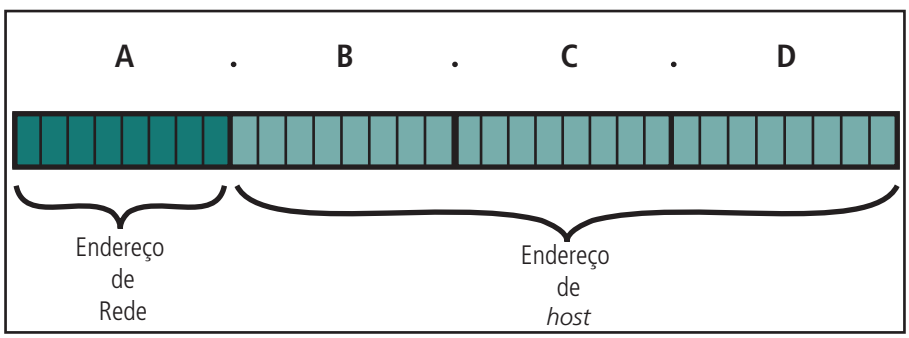


Figura 3.6: Formato do endereço classe A
Fonte: www.logicengenharia.com.br/mcamara/TPN/tpn_13.PDF

3.2.2 Redes classe B

Os endereços classe B utilizam o primeiro e o segundo octeto para endereço de rede, e os dois últimos octetos para endereço de *hosts*.

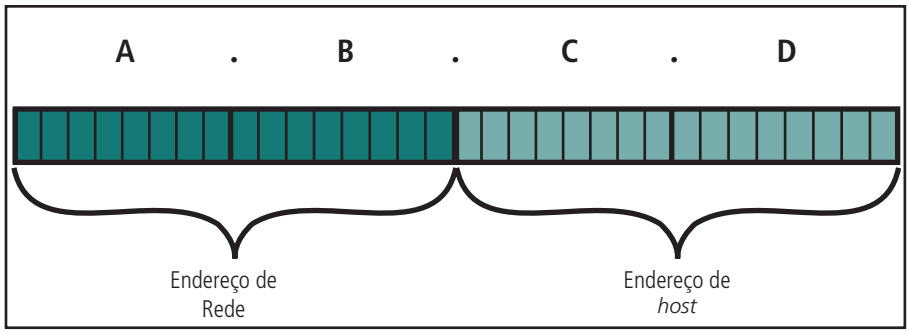


Figura 3.7: Formato do endereço classe B
Fonte: www.logicengenharia.com.br/mcamara/TPN/tpn_13.PDF

3.2.3 Redes classe C

As redes de classe C utilizam os três primeiros octetos para endereços de rede e o último octeto para endereço de *host*, porque o número de *hosts* que podem ser endereçados é igual a 254, variando de 1 a 254 nesse último octeto.

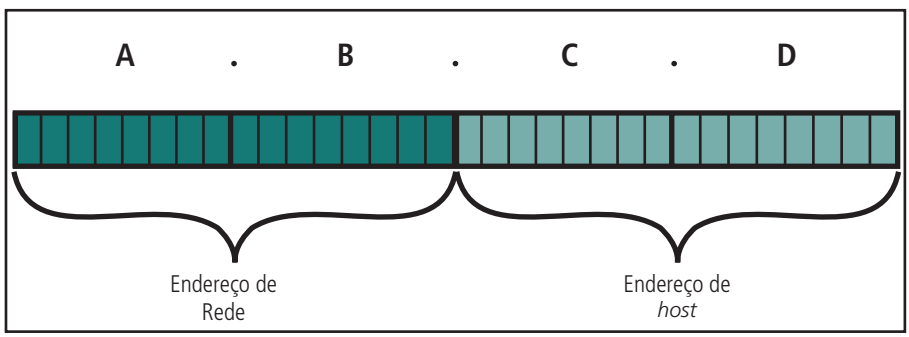


Figura 3.8: Formato do endereço classe C
Fonte: www.logicengenharia.com.br/mcamara/TPN/tpn_13.PDF

3.2.4 Redes classe D

A classe de endereçamento D é utilizada para envio de dados a um grupo específico de computadores, o que é chamado de *multicast*. Não é utilizada para endereçar computadores na rede.

Nessa classe, o valor do primeiro octeto pode variar de 224 a 239. Assim, os valores dos endereços podem variar de 224.0.0.0 a 239.255.255.255.

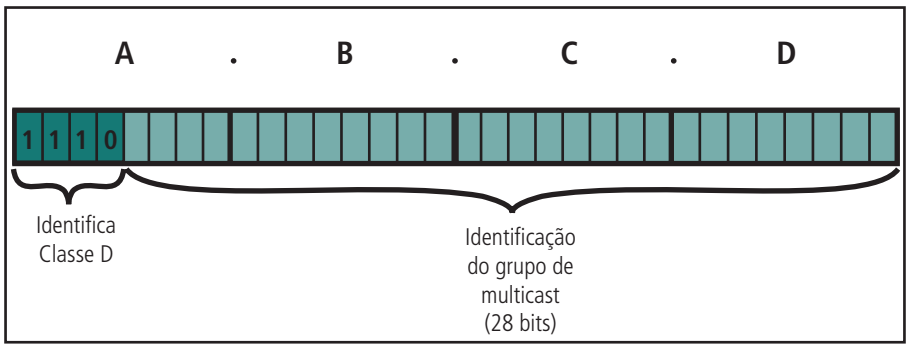


Figura 3.9: Formato do endereço classe D
Fonte: www.logicengenharia.com.br/mcamara/TPN/tpn_13.PDF

3.2.5 Redes classe E

A última classe de endereços IP é a classe E. Essa classe foi reservada para testes e uso futuro, não tendo qualquer aplicação convencional associada a ela. Mas, como o lançamento do endereçamento mais sofisticado, conhecido como IPv6, essa classe acabou se tornando inútil.

3.2.6 Endereço *loopback*

O IP define um endereço que é usado para testar os aplicativos de rede; este é o endereço de *loopback*. O IP reserva, na classe A, o prefixo 127 para *loopback*; se um *host* usa o prefixo 127, ele é "invisível para a rede". Por convenção, programadores costumam usar o endereço 127.0.0.1 para fazer o teste de *loopback*. Durante um teste, nenhum pacote pode ser enviado pelo *host*, consequentemente, o endereço de *loopback* nunca viaja pela rede (LIMA, 2012).

Quando um programa usa o endereço de *loopback, a* comunicação vai pelo caminho normal, saindo do nível de aplicação, passando pelo nível de transporte (TCP ou UDP) e chegando ao nível IP, que retorna a comunicação de volta ao nível de aplicação.

Tabela 3.2: Faixa de endereços decimais e binários por classe			
Classe	**Faixa de endereços**	**Representação binária**	**Utilização**
A	**1 – 126.X.X.X**	0nnnnnnn.hhhhhhhh. hhhhhhhh.hhhhhhhh	
B	**128 – 191.X.X.X**	10nnnnnn.nnnnnnnn. hhhhhhhh.hhhhhhhh	
C	**192 – 223.X.X.X**	1110xxxx.xxxxxxxx. xxxxxxxx.xxxxxxxx	
D	**224 – 239.X.X.X**	1110xxxx.xxxxxxxx. xxxxxxxx.xxxxxxxx	Multicast/Broadcast
E	**240 – 247.X.X.X**	11110xxx.xxxxxxxx. xxxxxxxx.xxxxxxxx	Reservado
Onde: **X é um número que varia de 0 a 250** **n é o número de *bits* da rede** **h é o número de *bits* do *host*** **x é o número de *bits* da rede e do *host***			

Fonte: http://www.wandreson.com/download/training-networking-tcpip.pdf

3.2.7 Endereços IP reservados

Assim como a classe de endereços 127.0.0.0 é reservada para *loopback*, existem alguns endereços, de acordo com Torres (2009), que são conhecidos como "endereços mágicos", que são endereços IP reservados para redes privadas ou outros endereços reservados que não podem ser utilizados em nenhuma máquina conectada à internet. E nenhum desses endereços pode ser anunciado. O que quer dizer que se uma máquina for conectada à internet com algum desses endereços reservado, ela não conseguirá passar pelos roteadores, sendo conhecida como rede privada.

Os endereços reservados estão mostrados na Figura 3.10 e são especificados pela RFC1597. RFC (*Request for Comments*), que é um documento que descreve os padrões de cada protocolo da internet.

Classe	Faixa de Endereço IP	No. De Redes
A	10 . 0 . 0 . 0	1 rede
B	172 . 16 . 0 . 0 a 172 . 31. 0 . 0	16 redes
C	192 . 168 . 0 . 0 a 192 . 168 . 255 . 0	256 redes

Figura 3.10: Faixas de endereços IP reservados
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-redes-ip/RedesIP-Cap-02.pdf, pag7

Para que as máquinas pertencentes à rede privada, que possuem endereços IP não válidos (ou reservados), possam navegar pela internet, é preciso converter esses seus endereço em endereços válidos. Isso pode ser feito pelo serviço co-

nhecido como NAT (*Network Address Translation*), que, também, é conhecido como *masquerading*. O aplicativo, ou serviço, NAT consiste em reescrever os endereços IP dos pacotes da rede privada, que passam pelo seu *gateway* padrão, em um único endereço válido, de maneira que um computador dessa rede tenha acesso à internet. O NAT será mais bem descrito na seção 3.8 desta aula.

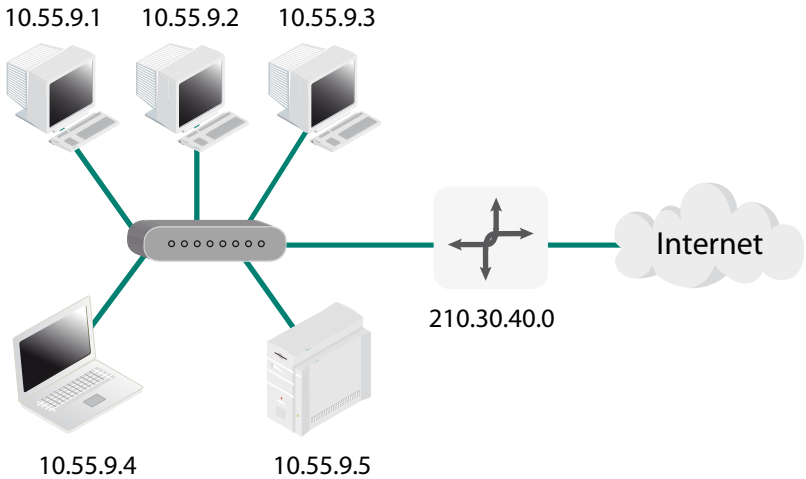


Figura 3.11: Exemplo de conexão de uma rede privada com a internet
Fonte: Coutinho (2010)

3.3 *BROADCAST* ou DIFUSÃO

Já sabemos que os endereços IP podem se referir tanto a *hosts* quanto a redes. E em geral o endereço terminado com 0 é considerado como *networkID* e terminado em 255 é considerado como endereço de *broadcast* (CISCONET, 2012). Assim nenhum endereço de *host* pode terminar em 0 ou 255.

Podemos concluir que endereços IP com os *bits* referentes ao HostID zerados irão se referir à rede propriamente dita, cuja numeração e classe são definidas pelo NetID. Já o endereço IP com os *bits* do HostID iguais a 1 significa referência a todos os *hosts* de sua NetID. Esse é então o endereço chamado de difusão ou endereço de *broadcast*. Ele permite a comunicação instantânea entre todos os *hosts* de um mesmo NetID.

Para enviar dados a todos os *hosts* pertencentes ao mesmo NetID, um *host* desse NetID precisa apenas enviar um único pacote com o endereço de *broadcast* dessa rede, que todos os outros *hosts* irão responder à solicitação simultaneamente.

Embora o endereço de *broadcast* oficial para uma rede seja o seu NetID mais seu HostID com todos os *bits* igual a 1, o endereço IP com todos os seus octetos iguais a 1, 255.255.255.255 é também tratado como *broadcast*.

Isso se refere a todos os *hosts* da rede local. É geralmente mais simples usar o 255.255.255.255 em vez de encontrar o número de rede para a rede local e formar um endereço de *broadcast*.

3.4 Endereço de *Multicast*

De acordo com Lima (2012), em algumas situações, o *broadcast* pode ser ineficiente. Isso ocorre, principalmente, quando é preciso enviar *frames* para alguns *hosts* na rede, mas não para todos. Pelo *broadcast* não conseguimos separar esses *hosts* e temos que enviar os *frames* para todos os *hosts* na rede. E mesmo que cada *host*, ou estação, na rede, possam ser configurados para descartar *frames* desnecessários, processar e descartar um *frame* requer recursos computacionais. Quando um *frame* chega, o *hardware* da interface de rede o coloca na memória, interrompe a CPU e permite que o *software* do sistema determine se o *fram* e deve ser ignorado. Assim, descartar *frames* envolve uma decisão da CPU. Temos então um grande desperdício de tempo de processamento em toda a rede.

Então, como computadores em uma LAN podem tirar vantagem do *broadcast* sem desperdiçar recursos da CPU? Para Lima (2012), a resposta está em uma forma restrita de *broadcast*, o *multicast*. Ele se parece muito com o *broadcast*, porém, quando os *frames* chegam, não são repassados diretamente para a CPU. A própria interface de rede faz a decisão de aceitar ou não o *frame*. Para que isso ocorra, o *hardware* deve ser programado com determinadas especificações, e só serão aceitos *frames* aos quais chegam essas especificações.

O endereço de *musticast* deverá pertencer à classe D. Assim, cada endereço entre 224.0.0.0 e 239.255.255.255 pode ser usado por um determinado grupo de *multicast*.

3.5 Notação decimal

Como já foi dito, os endereços IP são constituídos de *bits* e, para facilitar sua memorização, eles são usualmente representados em notação decimal. Na Tabela 3.3 pode-se ver a notação decimal do número 255.

Tabela 3.3: Exemplo de notação decimal do numero 255

Notação decimal do número 255							
1	1	1	1	1	1	1	1
128	64	32	16	8	4	2	1
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Fonte: Elaborado pelo autor

O sistema binário ou de base 2 é um sistema de numeração posicional em que todas as quantidades se representam com base em dois números, ou seja, zero e um (0 e 1). A combinação desses dois dígitos é que leva o computador a criar as várias informações da linguagem humana como letras, palavras, textos, números decimais, etc.

Exemplo de transformação de numeração binária para numeração decimal:

12(base10) = 1100(base2): 12 / 2 = 6 resto **0** – 6 / 2 = 3 resto **0** – 3 / 2 = 1 resto **1** – 1 / 2 = 0 resto **1**

15(base10) = 1111(base2) – Faça as contas e prove o resultado.

0(base10) = 0000(base2) – Faça as contas e prove o resultado.

Exemplo de transformação de numeração decimal para numeração binária:

1100(base2) = 12(base10): – 1x23 + 1x22 + 0x21 + 0x20 = 8 + 4 + 0 + 0 = 12

1111(base2) = 15(base10) – Faça as contas e prove o resultado.

0000(base2) = 0(base10) – Faça as contas e prove o resultado.

Pelos exemplos acima se pode constatar que com quatro dígitos binários consegue-se 16 combinações diferentes (de 0 (0000) a 15 (1111)). O que seria o mesmo que 24 é igual a 16, onde 2 é a base binária e 4 o número de *bits* a ser combinado

3.6 Máscaras de sub-redes

Um termo encontrado com facilidade ao configurar redes baseadas no protocolo TCP/IP é máscara de rede. A máscara de rede é formada por 32 *bits* no mesmo formato que o endereçamento IP, cada *bit* 1 da máscara informa a parte do endereço IP que é usada para o endereçamento da rede e cada *bit* informa a parte do endereço IP que é usada para o endereço das máquinas. Dessa forma, as máscaras padrões, também chamadas de *full*, são (TORRES, 2009):

- Classe A: 255.0.0.0.
- Classe B: 255.255.0.0.
- Classe C: 255.255.255.0.

Tabela 3.4: Exemplos de máscaras de sub-rede

Exemplo de Endereço IP	Classe do Endereço	Parte Referente à Rede	Parte referente ao *Host*	Máscara de Sub-rede Padrão
98.158.201.128	Classe A	98	158.201.128	255.0.0.0
158.208.189.45	Classe B	158.208	201.128	255.255.0.0
208.183.34.89	Classe C	208.183.34	89	255.255.255.0

Fonte: http://www.hardware.com.br/livros/linux-redes/entendendo-mascaras-sub-rede.html

Seja, por exemplo, o endereço IP 143.106.50.2, com a máscara 255.255.255.0; podemos extrair do endereço de rede (ID da rede) que é 143.106.50 e o endereço de *host* (ID *host*), que é 2.

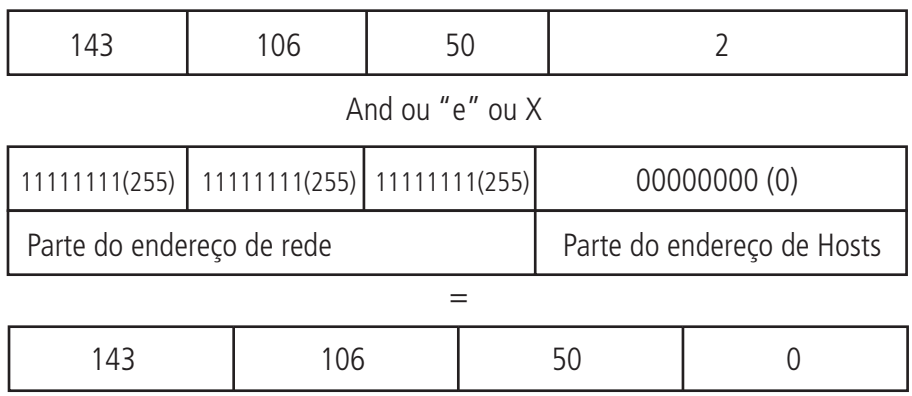


Figura 3.12: Exemplo de máscara de sub-rede
Fonte: Elaborada pelo autor

O exemplo da Figura 3.12 mostra uma configuração de um endereço IP com máscaras de sub-rede simples. Isso é o que chamamos de endereço *full*.

A máscara é usada fora de seus padrões *full* quando há a necessidade de segmentação da rede (TORRES, 2009). Isso é feito quebrando um octeto do endereço IP em duas partes, usando máscaras de sub-redes diferentes das máscaras *full*. Com isso tem-se dentro de um mesmo octeto uma parte que representa a rede e outra que representa o *host*. O intuito dessa configuração é formar sub-redes dentro de um mesmo endereço IP.

Uma sub-rede pode, então, ser definida como uma divisão lógica de uma rede IP, classe *full*, em redes menores dentro da mesma faixa desse endereço IP. Cria-se, com isso, a possibilidade de partilhar um mesmo endereço IP entre diversas sub-redes. O importante, nesse caso, é que todos os computadores pertencentes a uma mesma sub-rede da rede sejam configurados com a mesma máscara; senão eles poderão não se comunicar, pois, logicamente, estarão conectados a redes diferentes. Para esse tipo de configuração é necessário configurar o endereço IP usando números binários e não decimais. Como já foi visto, o número decimal 255 (equivalente a 11111111) indica que todos os oito números binários do octeto se referem à rede, enquanto o decimal 0 (correspondente a 00000000) indica que todos os oito binários do octeto se referem ao *host*.

Seja um endereço IP 200.131.47.X , que, como já foi visto anteriormente, é um endereço IP classe C, onde 200.131.47 é o endereço da rede e o X o endereço do *host*. Pretende-se, por exemplo, dividi-lo em duas sub-redes distintas. O problema é que é impossível alterar o endereço 200.131.47, que já representa o ID da rede. O caminho a ser tomado é usar uma máscara de sub-rede diferente do padrão, que nesse caso da classe C, é 255.255.255.0.

Decimal:	**200**	**131**	**47**	**X**
Binário:	**11001000**	**10000011**	**00101111**	**????????**
Máscara:	**255**	**255**	**255**	**0**
Binário:	**11111111**	**11111111**	**11111111**	**00000000**
	Rede	**Rede**	**Rede**	**Host**

Figura 3.13: Representação do IP 200.131.47.X com máscara 255.255.255.0 em binário
Fonte: Elaborada pelo autor

Para resolver esse problema, é necessário reservar parte dos oito *bits* do último octeto da máscara de sub-rede desse endereço IP (classe C) para o endereçamento da redeID e o restante endereça o *host* ID. Dessa maneira, uma parte desses oito *bits* será 1, que como foi dito, refere-se à rede, e a outra parte será 0, representando o hostID. Para esse exemplo de duas sub-redes, a máscara, que em vez de ser 255.255.255.0, corresponde ao binário (11

111111.111111.11111111.00000000), deverá ser 255.255.255.192 (corresponde ao binário 11111111.111111.11111111.11000000). Isso significa que os dois primeiros *bits* do último octeto referenciam a rede, pois são 1, e os seis últimos referenciam o *host*, pois são 0.

Decimal:	200	131	47	X
Binário:	11001000	10000011	00101111	????????
Máscara:	255	255	255	192
Binário:	11111111	11111111	11111111	11000000
	Rede	Rede	Rede	Rede Host

Figura 3.14: Representação do IP 200.131.47.X com máscara 255.255.255.240 em binário
Fonte: Elaborada pelo autor

Então, nessa nova configuração, o último octeto foi dividido em dois endereços binários, um com dois *bits* 1, que endereça as sub-redes e outro com seis *bits* 0, que endereça os *hosts*. Essa configuração é feita através de potência de 2, já que se trabalha com número binário, chegando ao resultado que essa cominação possibilita, que são quatro sub-redes logicamente separadas dentro de um mesmo endereço IP (22 = 4).

Em um endereço IP *full*, perde-se o primeiro e o último endereço, pois o primeiro é o endereço de rede e o último, o endereço de *broadcast*. Baseado nisso, pode-se afirmar que no endereço 200.131.47.X do exemplo, tomando-o apenas como um classe C *full*, o endereço 200.131.47.0 é o endereço de rede e não é usado para endereçar *host*. Da mesma maneira, o endereço 200.131.47.255 é o endereço de *broadcast*, também não podendo endereçar *host*. A mesma definição leva-se para as sub-redes, onde a primeira e a ultima sub-rede não podem ser usadas.

Primeiramente é necessário observar que, usando a máscara de rede 255.255.255.192, o endereço 200.131.47.X não é mais um endereço classe C *full*, pois foi dividido em quatro sub-redes. E trazendo a mesma definição de se perder o primeiro e o último endereço IP *full*, pode-se afirmar que a primeira sub-rede e a última são perdidas, assim como o primeiro e o último endereço de cada sub-rede.

Um endereço classe C *full* (máscara de sub-rede 255.255.255.0, onde o último octeto é variável para endereçar *hosts*), possibilita formar 256 endereços diferentes (28 = 256). Tomando como base o endereço 200.131.47.X classe C *full*, pode-se constatar que o primeiro endereço é 200.131.47.0 (X=0), e o último igual a 200.131.47.255 (X=255), o que formam 256 combinações

diferentes. Mas lembre-se que o primeiro e o último endereços são perdidos; então, o primeiro endereço válido para *host* é 200.131.47.1 e o último, 200.131.47.254.

Agora vamos definir quantos *hosts* pode-se ter em cada sub-rede para a máscara 255.255.255.192. Foi visto que essa máscara de sub-rede possibilita a formação de quatro sub-redes distintas. Então, basta pegar a quantidade de endereços da classe *full* e dividir por 4 (256/4=64). Chega-se ao resultado de 64 endereços por sub-rede:

- primeira sub-rede vai de 200.131.47.0 até 200.131.47.63;
- segunda sub-rede vai de 200.131.47.64 até 200.131.47.127;
- terceira sub-rede vai de 200.131.47.128 até 200.131.47.191;
- quarta sub-rede vai de 200.131.47.192 até 200.131.47.255.

Como se perdem a primeira e a última sub-rede e o primeiro e o último endereço de cada sub redes, chega-se ao seguinte resultado dos endereços válidos:

- primeira sub-rede vai de 200.131.47.65 até 200.131.47.126;
- segunda sub-rede vai de 200.131.47.129 até 200.131.47.190.

Dos endereços das sub-redes acima, pode-se constatar que, por exemplo, os *hosts* 200.131.47.70 e 200.131.47.140 não pertencem à mesma sub-rede; portanto, não se comunicam diretamente, sendo necessária a existência de um roteador entre eles. E por último, lembre-se que o endereço da primeira sub-rede válida é 200.131.47.64 e seu endereço de broadcast é 200.131.47.127; e da segunda sub-rede válida é 200.131.47.128 e seu endereço de broadcast é 200.131.47.191. Esse processo pode ser feito para qualquer classe de endereço IP *full*.

As vantagens em usar sub-redes são:

- dividir redes grandes em redes menores;
- conectar diferentes redes físicas. As redes tornam-se sub-redes de uma rede maior conectadas por roteadores;

- distinguir redes locais;

- isolar partes da rede. Pode-se querer restringir o tráfego em uma sub-rede, por motivo de segurança ou diminuição de tráfego.

A maior e talvez única desvantagem é a grande perda de endereços IP, o que torna seu uso bastante limitado.

3.6.1 Tabelas de máscara de sub-rede decimal *versus* binária por classes IP

Tabela 3.5: Conversão da máscara de sub-rede para redes de classe A

N. de sub-redes	N. de *bits*	Máscara de sub-rede	N. de *hosts* por sub-rede
0	1	*Invalid*	*Invalid*
2	2	255.192.0.0	4,194,302
6	3	255.224.0.0	2.097,150
14	4	255.240.0.0	1,048,574
30	5	255.248.0.0	524,286
62	6	255.252.0.0	262.142
126	7	255.254.0.0	131,070
254	8	255.255.0.0	65,534

Fonte: Elaborada pelo autor

Tabela 3.6: Conversão da máscara de sub-rede para redes de classe B

N. de sub-redes	N. de *bits*	Máscara de sub-rede	N. de *hosts* por sub-rede
0	1	*Invalid*	*Invalid*
2	2	255.255.192.0	16.282
6	3	255.255.224.0	8.190
14	4	255.255.240.0	4.094
30	5	255.255.248.0	2.046
62	6	255.255.252.0	1.022
126	7	255.255.254.0	510
254	8	25.255.255.0	254

Fonte: Elaborada pelo autor

Tabela 3.7: Conversão da máscara de sub-rede para redes de classe C

N. de sub-redes	N. de *bits*	Máscara de sub-rede	N. de *hosts* por sub-rede
0	1	*Invalid*	*Invalid*
2	2	255.255.255.192	64
6	3	255.255.255.224	32
14	4	255.255.255.240	16
30	5	255.255.255.248	8
62	6	255.255.255.252	4

Fonte: Elaborada pelo autor

Pode-se concluir que no IPv4 uma rede ou sub-rede é identificada por seu endereço IP base e sua máscara de sub-rede.

A base do endereço IP que era constituída por: <endereço de rede> <endereço de host> passa a ser interpretada como: <endereço de rede> <endereço de máscara de sub-rede> <endereço de host>.

3.7 DHCP (*Dynamic Host Configuration Protocol*)

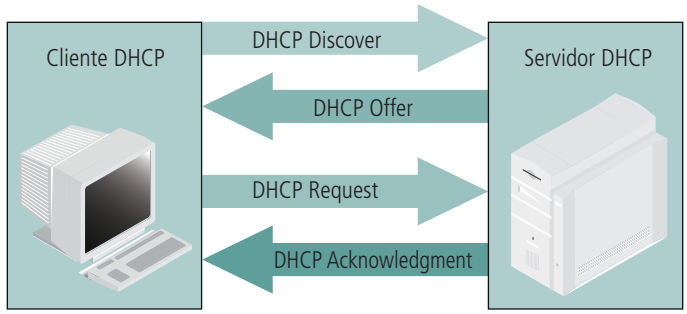
DHCP, abreviatura de *Dynamic Host Configuration Protocol*, é um serviço utilizado para automatizar as configurações do protocolo TCP/IP nos dispositivos de rede (computadores, impressoras, *hubs*, *switches*, ou seja, qualquer dispositivo conectado à rede e que esteja utilizando o protocolo TCP/IP).

Sem o uso do DHCP, o administrador da rede e a sua equipe teriam que configurar, manualmente, as propriedades do protocolo TCP/IP em cada dispositivo de rede (genericamente denominados *hosts*). Com o uso do DHCP, essa tarefa pode ser completamente automatizada. O uso do DHCP traz diversos benefícios, dentre os quais podemos destacar os seguintes (BATTISTI, 2012):

- Automação do processo de configuração do protocolo TCP/IP nos dispositivos da rede.

- Facilidade de alteração de parâmetros tais como *default gateway*, servidor DNS e assim por diante, em todos os dispositivos da rede, através de uma simples alteração no servidor DHCP.

- Eliminação de erros de configuração, tais como digitação incorreta de uma máscara de sub-rede ou utilização do mesmo número IP em dois dispositivos diferentes, gerando um conflito de endereço IP.



1 - O cliente de DHCP pede um endereço de IP *(DHCP Discover)*
2 - É oferecido um endereço *(DHCP Offer)* pelo servidor
3 - O cliente aceita a oferta do endereço *(DHCP Request)*
4 - É nomeado o endereço oficialmente *(DHCP Acknowledgement)*.

Figura 3.15: Funcionamento do DHCP
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-redes-ip/RedesIP-Cap-02.pdf.

Além do servidor DHCP, os computadores da rede devem possuir um *software* cliente DHCP para se comunicar com esse servidor. Esse *software* cliente DHCP solicita e obtém do servidor DHCP as configurações básicas do TCP/IP, quando ele é ligado. Essa alocação dinâmica e feita a partir de uma faixa de endereços IP configuradas para esse fim.

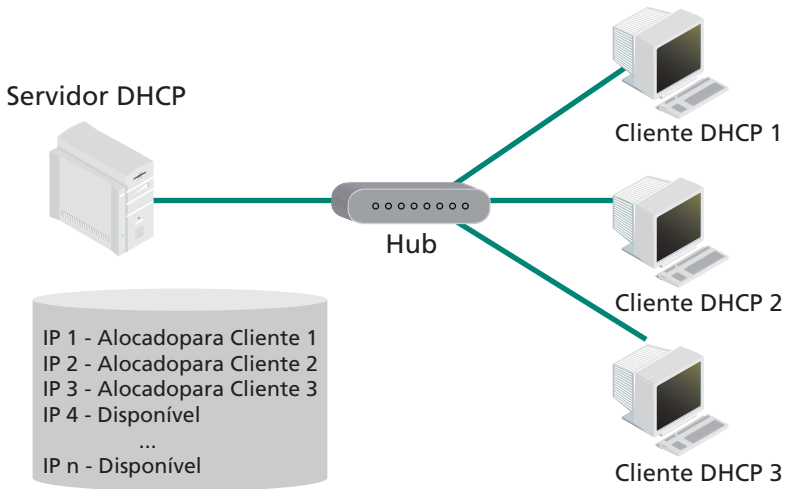


Figura 3.16: Exemplo de uma rede utilizando servidor DHCP
Fonte: Coutinho (2010)

3.8 NAT (*Network Address Translation*)

Quando uma LAN é conectada à internet, é preciso que todos os computadores tenham endereços válidos para se comunicar com ela. Mas, com o surgimento das redes LAN privadas, surgiu o problema de como os computadores pertencentes a essas redes privadas poderiam receber as respostas aos seus pedidos feitos para a internet, que, por se tratarem de uma rede privada,

seu endereços IP nunca poderiam ser passados para a internet porque não são roteados nela. Dessa maneira, um computador externo que recebesse um pedido de um endereço IP provado não saberia para onde enviar a resposta. A solução seria gerar os pedidos através de um IP global de um roteador ou *gateway* padrão. E quando a resposta chegasse ao roteador ou ao *gateway* padrão, seria preciso saber a qual dos computadores presentes na LAN pertencia aquela resposta. Uma solução para esse problema é fazer um mapeamento no roteador ou no *gateway* padrão, através do serviço chamado de NAT.

Com o uso do NAT (*Network Address Translation*), também conhecido como *masquerading*, os computadores da rede interna utilizam os chamados endereços privados. Os endereços privados não são válidos na internet, isto é, pacotes que tenham como origem ou como destino um endereço na faixa dos endereços privados não serão encaminhados, serão descartados pelos roteadores. O *software* dos roteadores está configurado para descartar pacotes com origem ou destino dentro das faixas de endereços IP privados. As faixas de endereços privados são definidas na RFC 1597 e estão indicadas a seguir (BATTISTI, 2012):

- 10.0.0.0 / 10.255.255.255
- 172.16.0.0 / 172.31.255.255
- 192.168.0.0 / 192.168.255.255

O mapeamento que o serviço NAT (instalado no roteador ou *gateway* padrão) faz é baseado no IP interno do computador que quer enviar uma requisição para a internet conjuntamente com a porta que ele destinou para gerenciar essa requisição. Esse mapeamento é feito através da geração de um número de 16 *bits*, baseado nesses dois parâmetros do computador local, pelo servidor NAT, e é gerada uma tabela no servidor NAT que contém a relação endereço IP mais porta de origem do computador local com o numero de 16 *bits* gerado. Após, a requisição do computador local é enviada para a internet através do par endereço IP válido, mais porta de origem do servidor NAT, a qual recebe o número de 16 *bits* gerado por ele. Quando o computador remoto receber o pedido, ele sabe a origem da requisição, que não é o computador local, mas sim o servidor NAT. Quando o servidor NAT recebe a resposta do computador remoto, ele faz a operação inversa, procurando na sua tabela a entrada que corresponda ao endereço IP mais a porta do computador local. Ao encontrar a entrada, é feito o direcionamento para o computador correto dentro da rede privada.

Resumindo, o servidor NAT tem a função de traduzir os endereços válidos e registrados de acesso à internet para os endereços reservados da rede interna e vice-versa. Esse serviço de tradução dos endereços é sempre implementado em um roteador ou *gateway* padrão, ou seja, na entrada e saída da rede local. Normalmente, junto com o serviço NAT é instalado um serviço de *firewall*, que tem a finalidade de bloquear acessos não autorizados aos recursos do sistema ou da rede. O uso do NAT possibilitou, também, a economia de endereços IP, já que vários computadores de uma rede privada podem acessar a internet através de um único endereço IP válido.

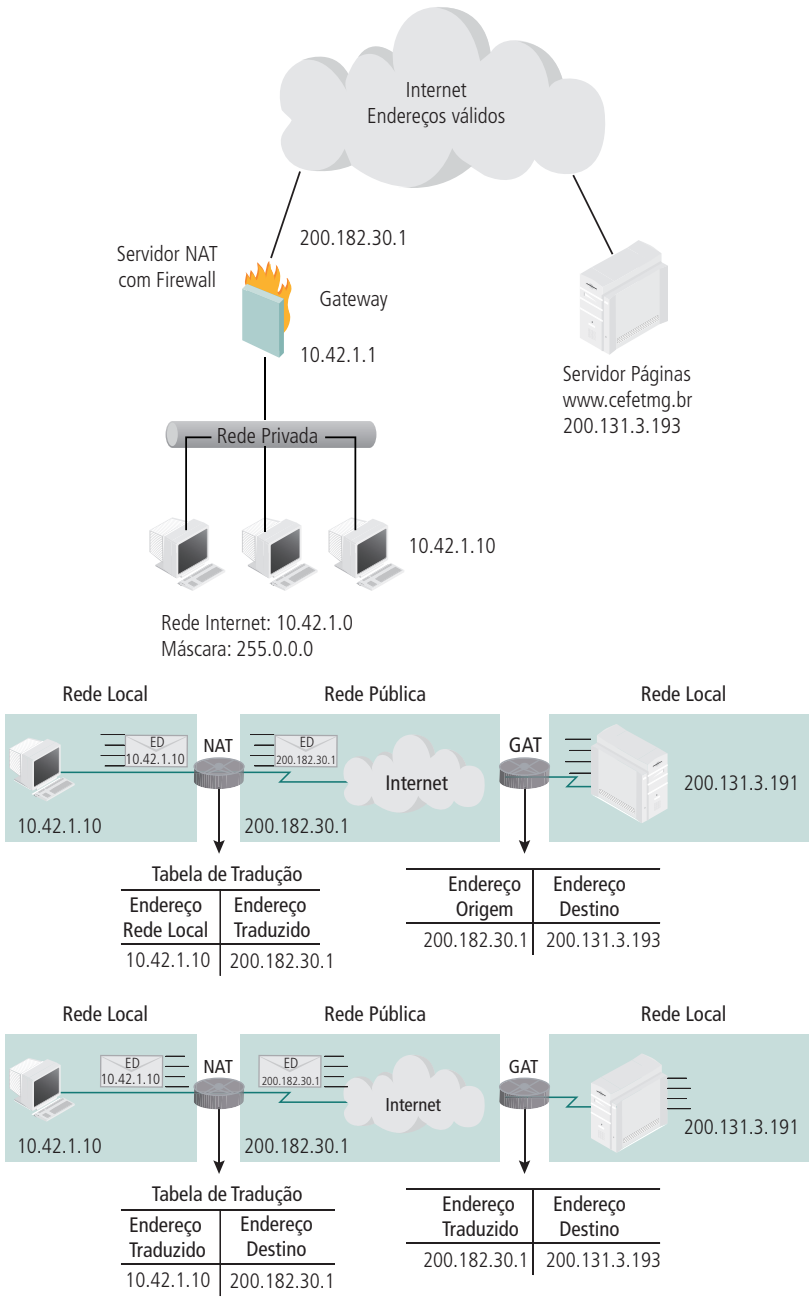


Figura 3.17: Funcionamento do serviço NAT
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-redes-ip/RedesIP-Cap-02.pdf

3.9 Problemas no uso de endereçamento IP

Um primeiro problema do esquema de endereçamento IP é que, como ele se refere a uma conexão de rede (e não a um *host*), quando uma máquina muda de uma rede para outra, ela deve mudar de endereço IP. Isso traz uma grande barreira à conexão de *hosts* móveis (como computadores portáteis) que podem precisar de IPs fixos à internet.

Um segundo problema está na perda de endereços IP. Para exemplificar, se uma rede local possui apenas dez *hosts*, ela deve receber um endereço IP classe C, que é a menor classe que pode atendê-la. Nesse caso, haverá um desperdício de 244 endereços IP (254 – 10).

Um terceiro problema, talvez menos importante, está no caso de uma rede LAN endereçada com IPs classe C que cresça para além de 255 *hosts*. Nesse caso ela deve ser realocada para a classe B, ou ser dividida em duas classes C. O que, nos dois casos, implica uma mudança de estrutura.

Um quarto problema, e talvez o maior deles, surge quando analisamos cuidadosamente uma situação especial de roteamento de pacotes na internet. Já dissemos que as decisões de roteamento dependem da extração do NetID. Para exemplificar, considere uma máquina conectada a duas redes (Rede 1 e Rede 2). Como o roteamento de pacotes para esse *host* será determinado pelo seu NetID (e ele possui dois distintos), o caminho tomado por um pacote que se destina a essa máquina irá depender do endereço usado pelo remetente. Assim, parâmetros como o tempo de resposta na comunicação irão variar de acordo com a interface que será endereçada. Essa multiplicidade de caminhos pode trazer consequências pouco óbvias. Um *host* pode deixar de ser acessível por um de seus endereços IP, caso haja algum impedimento físico em uma das redes a que ele está conectado. Outra máquina que conheça apenas esse endereço desativado e se comunique com esse *host* através dele não poderá mais fazê-lo, embora o *host* ainda esteja ligado à internet.

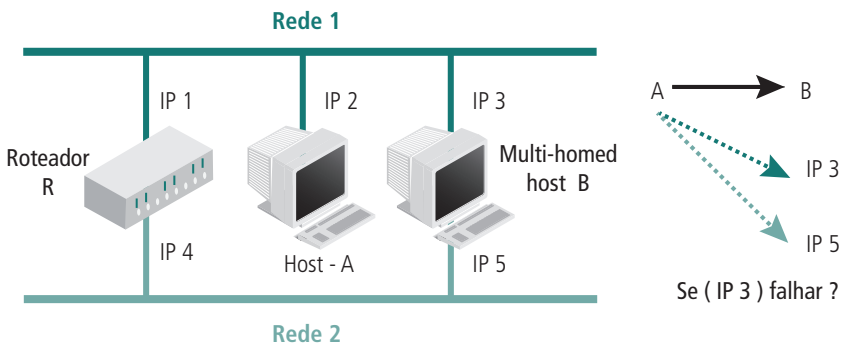


Figura 3.18: Problema da multiplicidade de caminhos
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-redes-ip/RedesIP-Cap-02.pdf

Fixe seus conhecimentos sobre o endereço IPv4 assistindo ao vídeo disponível em http://www.youtube.com/watch?v=EG9mSXlMTU4&feature=related

Para saber mais sobre a função da máscara de sub-rede, acesse http://www.vivaolinux.com.br/artigo/Calculando-mascara-de-subrede-e-broadcast

3.10 Notação CIDR

Como já foi explicado, para que uma rede LAN possa trabalhar com a suíte TCP/IP, ela necessita de endereços IPs em seus computadores. Esses endereços são compostos de dois parâmetros, que são o endereço IP conjuntamente com a máscara de sub-rede. Foi visto também que, ao utilizar esse par de parâmetros para construir sub-redes, tem-se uma perda considerável de endereços IP.

Para Comer (2007), uma técnica conhecida como CIDR (de *Classless Inter-Domain Routing*) resolve o problema. Conceitualmente o CIDR reduz um bloco de endereçamentos IP contínuos (por exemplo, cinco endereços classe C contínuos) a uma única entrada representada pelo par "endereço de rede, contagem", o qual é o menor endereço de rede do bloco, e contagem especifica o seu número total. Essa notação é chamada de *standard*.

Por exemplo, o par "192.5.48.0, 3" pode ser usado para especificar os três endereços de rede "192.5.48.0, 192.5.49.0, 192.5.50.0" (COMER, 2007).

3.10.1 Notação *standard*

Como vimos , a notação *standard* é escrita começando com o endereço de rede, na direita com o número apropriado de *bits* com valor zero. Esse endereço é seguido, normalmente, pelo caráter "/" e o comprimento de um prefixo, em *bits*, definindo o tamanho da rede (o prefixo é, na verdade, o comprimento da máscara de sub-rede).

Como exemplo prático, veja o endereço IP na notação *standard* CIDR 192.168.0.0 / 24. O /24 tem de ser escrito em quatro octetos, pois é o padrão IPv4, ou seja 32 *bits*. Contando os 24 *bits* da esquerda para direita, e preenchendo o resto com zero para completar 32 *bits*, temos o numero binário: 11111111.11111111.11111111.00000000 (8 + 8 +8 +0 =24). Como esse prefixo substitui a máscara de sub-rede, é como se ele fosse 255.255.25.0; então, esse endereço representado em IP classes seria um endereço classe C 192.168.0.0 que possui máscara de sub-rede 255.255.255.0.

Agora veja o endereço IP na notação *standard* CIDR 192.168.0.0 / 22. Contando os 22 *bits* da esquerda para a direita, e preenchendo o resto com zero para completar 32 *bits*, tem-se que o /22 é 11111111.11111111.11111100.00000000 (8 + 8 + 6 + 0). Em classe *full* será escrito como 192.168.0.0 e máscara de sub-rede 255.255.252.0, pois a transformação de 11111100 para decimal é 252. Faça as contas e comprove.

Fazendo a correlação do endereçamento CIDR com o endereço em classes, pode-se fazer as seguintes observações:

- A Classe A inicia-se com "0" como o bit mais significativo. Os próximos sete *bits* de um endereço de Classe A identificam a rede (o primeiro octeto) e os restantes 24 *bits* são usados para endereçar o *host*. Portanto, a máscara de sub-rede da classe A é 11111111.0.0.0. A notação CIDR nesse caso é /8. Então a notação CIDR para uma rede de Classe A *full* é "/8".

- A Classe B inicia-se com "10" como os *bits* mais significativos. Os próximos 14 *bits* identificam a rede (os dois primeiros octetos) e os 16 *bits* restantes servem para endereçar os *hosts*. Portanto, a máscara de sub-rede da classe B é 11111111.11111111.0.0. A notação CIDR nesse caso é /16. Então a notação CIDR para uma rede de Classe B *full* é "/16".

- A Classe C é identificada pelos *bits* mais significativos (chamados também MSB, *Most Significant Bits*) iguais a "110". Os próximos 21 *bits* identificam a rede (os três primeiros octetos) e os oito *bits* restantes servem para endereçar os *hosts*. Portanto, a máscara de sub-rede da classe C é 11111111.11111111.11111111.0. A notação CIDR nesse caso é /24. Então a notação CIDR para uma rede de Classe C *full* é "/24".

- Existe também uma série de endereços que se iniciam com "111", e que são muito usados para outros fins (por exemplo, *multicast*) e que não nos interessam aqui.

IPv4 CIDR Chart			RIPE NCC
IP Addresses	**Bits**	**Prefix**	**Subnet Mask**
1	0	/32	255.255.255.255
2	1	/31	255.255.255.254
4	2	/30	255.255.255.252
8	3	/29	255.255.255.248
16	4	/28	255.255.255.240
32	5	/27	255.255.255.224
64	6	/26	255.255.255.192
128	7	/25	255.255.255.128
256	8	/24	255.255.255.0
512	9	/23	255.255.254.0
1 K	10	/22	255.255.252.0
2 K	11	/21	255.255.248.0
4 K	12	/20	255.255.240.0
8 K	13	/19	255.255.224.0
16 K	14	/18	255.255.192.0
32 K	15	/17	255.255.128.0
64 K	16	/16	255.255.0.0
128 K	17	/15	255.254.0.0
256 K	18	/14	255.252.0.0
512 K	19	/13	255.248.0.0
1 M	20	/12	255.240.0.0
2 M	21	/11	255.224.0.0
4 M	22	/10	255.192.0.0
8 M	23	/9	255.128.0.0
16 M	24	/8	255.0.0.0
32 M	25	/7	254.0.0.0
64 M	26	/6	252.0.0.0
128 M	27	/5	248.0.0.0
256 M	28	/4	240.0.0.0
512 M	29	/3	224.0.0.0
1024 M	30	/2	192.0.0.0
2048 M	31	/1	128.0.0.0
4096 M	32	/0	0.0.0.0
Contact Registration Services: **hostmaster@ripe.net - lir-help@ripe.net**			**www.ripe.net**

K = 1,024 . M = 1,048,576

Figura 3.19: Tabela de consulta da notação CIDR
Fonte: http://snnangola.files.wordpress.com/2009/02/cidr_working.png

3.10.2 Agregação de prefixos de roteamento

Talvez o benefício mais importante da notação CIDR seja a possibilidade de agregação de prefixos de roteamento. A agregação de rotas permite que uma única entrada na tabela de rotas possa representar várias sub-redes (PEREIRA, 2012). Por exemplo, 16 redes /24 contíguas podem agora ser agregadas e mostradas como uma rota única de /20 (caso os primeiros 20 *bits* dos endereços de rede coincidam). Dois /20 contíguos podem ser agregados num /19, e assim por diante. Isto permite uma redução significativa do número de rotas, prevenindo um aumento significativo das tabelas de roteamento. Assim, pode-se usar várias faixas de endereços contínuas para que sejam agrupadas em faixas maiores, de forma a simplificar a configuração.

Para um exemplo prático, imagine uma rede local com mais de 254 *host*, 400 *hosts*, por exemplo; ela necessitaria ou de um endereço classe B, que é muito difícil de conseguir, ou dois endereços classe C. No caso de dois endereços da classe C, seria necessário existir um roteador entre eles. Com o CIDR é possível agrupar as faixas de endereços classe C contínuas em uma única faixa com máscara /21, por exemplo, que oferece um total de 510 (255+255, pois ainda continua sendo necessário separar o endereço da rede e o endereço de *broadcast* dos endereços de *host*). O endereço formado pelas duas sub-redes é comumente chamado de super-rede.

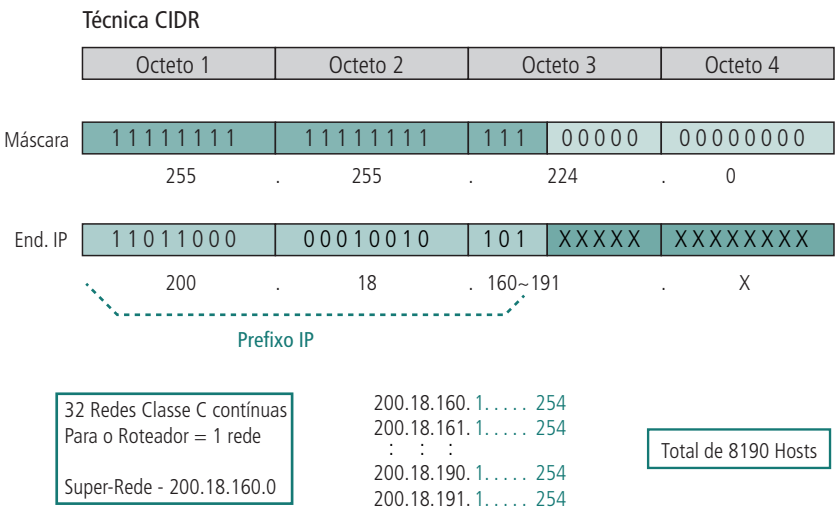


Figura 3.20 Exemplo de uma super-rede
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-redes-ip/RedesIP-Cap-02.pdf

3.11 ENDEREÇAMENTO VLSM (*VARIABLE LENGTH SUBNET MASK*)

Para Filippetti (2008), VLSM consiste em segmentar sub-redes em blocos não necessariamente do mesmo tamanho. Daí o nome "sub-redes de tamanho variável". Dessa maneira, a alocação dos endereços IP em sub-redes é feita de acordo com as necessidades individuais e não nas regras de uso das máscaras fixas usadas quando se trabalha com endereços IP em classes. Assim a divisão de rede/*host* pode ocorrer em qualquer fronteira de *bits* no endereço, ou seja, em blocos não necessariamente do mesmo tamanho. Porque as distinções de classes normais são ignoradas, o novo sistema foi chamado de roteamento sem classes.

Uma vantagem em usar o VLSM está na diminuição da tabela de roteamento, pois ele permite a sumarização de múltiplas pequenas sub-redes, geradas por grandes redes, dentro de uma única grande rota.

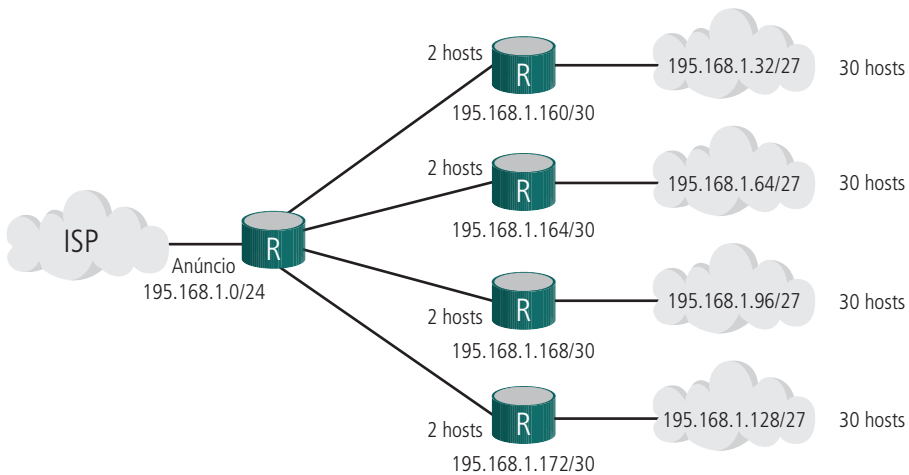


Figura 3.21: Exemplo de endereçamento VLSM
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-redes-ip/RedesIP-Cap-02.pdf

3.12 Diferença entre CIDR e VLSM

De acordo com Pereira (2012), CIDR e VLSM são essencialmente a mesma coisa, pois ambos permitem dividir recursivamente uma porção do espaço de endereços IP em pedaços (blocos) menores.

A diferença é que com VLSM a recursão é feita no espaço de endereçamento previamente alocado para a organização, sendo isso invisível para a internet global. O CIDR, por sua vez, permite a alocação recursiva de um bloco de endereços por um internet *Registry* a um "*high-level* ISP ", a um "*middle-level* ISP", a um "*low-level* ISP" e, finalmente, à rede privada da organização (PEREIRA, 2012).

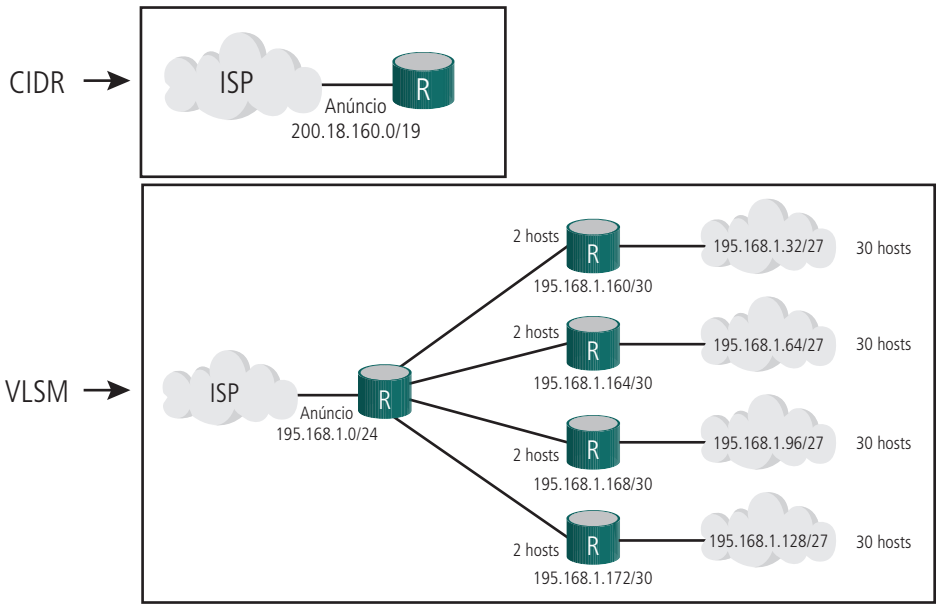


Figura 3.22: CIDR X VLSM
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-redes-ip/RedesIP-Cap-02.pdf

3.13 IPv6

Quando a internet surgiu, a versão do protocolo IP usada era a 4; por isso esses endereços são chamados de IPv4. Desde então o poder de processamento das máquinas aumentou substancialmente, assim como a quantidade delas conectadas à internet. O IPv4 vem conseguindo acomodar todas essas mudanças, embora não tenha sido originalmente projetado para dar suporte a uma rede de escala universal ou para permitir aplicações multimídia. A necessidade de um *upgrade* começa a aparecer aqui.

Segundo Kurose (2010), no começo da década de 1990, a Internet Engineering Task Force (IETF) iniciou um esforço para desenvolver o sucessor do protocolo IPv4. A motivação para isso foi a preocupação de que o espaço de 32 *bits* estava começando a escassear.

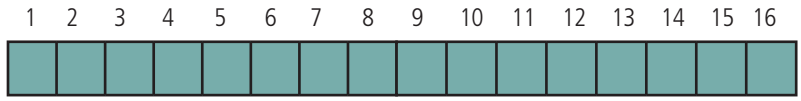
O servidor NAT, a adoção do CIDR e o VLSM abrandaram um pouco essa preocupação. Mas essas soluções eram paliativas.

Assim, o IETF, desenvolveu o IPv6. Essa nova versão foi desenvolvida tendo em mente que deveria ser um passo evolucionário em relação à versão 4. Assim, funções desnecessárias do IPv4 foram removidas, e funções que trabalhavam bem foram mantidas; e novas funcionalidades foram acrescentadas. Além do que, o IPv6 foi projetado para facilitar a migração do IPv4. Dessa maneira o IPv6 está sendo implantado gradativamente na internet.

De acordo com Albuquerque (2001), "o protocolo foi desenvolvido para atender não apenas às necessidades atuais, mas também às necessidades das aplicações futuras".

De acordo com Kurose (2010), a base do protocolo IPv6 está no seu espaço de endereçamento, que foi aumentado de 32 para 128 bits. Essa ampliação é uma das mais importantes características do novo protocolo. Agora a quantidade de endereços é 2128, ou seja, um imenso espaço de endereçamento. Para endereçar essa quantidade enorme de endereços, o IPv6 utiliza oito sequências de até quatro caracteres separado por ':' (sinal de dois pontos), mas considerando o sistema hexadecimal, pois ficaria inviável a sua representação em binário ou em decimal, como é feito no IPv4. Com essa quantidade de endereços, o IPv6 acaba com as classes de endereços.

Endereço no IP v6:



Notações:

Binário: impraticável (128 bits)
Decimal com ponto: 104 . 230 . 140 . 33 . 87 . 255 . 34 . 0 . 17 . 0 . 0 . 123 . 255 . 255 . 255
Hexadecimal com dois pontos: 6675 : 9C8A : FFFF : FFFF : 0 : 1180 : FFFF : 196A

Figura 3.23: Formas de endereçamento IPv6
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-redes-ip/RedesIP-Cap-02.pdf

O *header* (ou cabeçalho) do IPv6 foi simplificado, de modo a reduzir o seu processamento. Tanto é que, apesar de os endereços da versão 6 serem quatro vezes maiores que os da versão 4, seu cabeçalho é duas vezes maior. Isso gera ganho de tempo na leitura do IPv6

A flexibilidade de inclusão de opções no cabeçalho do IPv6 foi outra facilidade implementada, para que novas extensões fossem incluídas. Isso possibilita a inclusão de elementos como os que fornecem suporte para autenticação, integridade de dados e confidencialidade. Essa característica faz o protocolo IPv6 ser bem mais seguro que o IPv4.

Uma nova capacidade foi adicionada para permitir que o transmissor de um dado pacote possa requerer um fluxo especial para ele. Essa capacidade faz com que a qualidade da transmissão seja melhorada, podendo ser priorizadas aplicações que têm uma transmissão contínua em relação a outras que não têm esse fluxo.

3.13.1 Representação dos endereços IPv6

A representação dos endereços IPv6 divide o endereço em oito grupos de 16 *bits*, separando-os por ":", escritos com dígitos hexadecimais (0-F). Por exemplo (SANTOS, 2010):

- 2001:0DB8:AD1F:25E2:CADE:CAFE:F0CA:84C1

Ainda de acordo com Santos (2010), o IPv6 aceita tanto caracteres maiúsculos quanto minúsculos. E regras de abreviação podem ser aplicadas para facilitar a escrita de alguns endereços muito extensos. É permitido omitir os zeros à esquerda de cada bloco de 16 *bits*, além de substituir uma sequência longa de zeros por "::".

Por exemplo, o endereço 2001:0DB8:0000:0000:130F:0000:0000:140B pode ser escrito como 2001:DB8:0:0:130F::140B ou 2001:DB8::130F:0:0:140B. Neste exemplo é possível observar que a abreviação do grupo de zeros só pode ser realizada uma única vez; caso contrário, poderá haver ambiguidades na representação do endereço. Se o endereço acima fosse escrito como 2001:DB8::130F::140B, não seria possível determinar se ele corresponde a 2001:DB8:0:0:130F:0:0:140B, a 2001:DB8:0:0:0:130F:0:140B ou 2001:DB8:0:130F:0:0:0:140B (SANTOS, 2010).

A abreviação pode ser feita no fim ou no início do endereço. Santos (2010) cita como exemplo o 2001:DB8:0:54:0:0:0:0, que pode ser escrito da forma 2001:DB8:0:54::.

Uma parte importante do endereço IP é a parte que determina o prefixo da rede. Nos endereços IPv6 ela continua sendo escrita do mesmo modo que no IPv4, que utiliza a notação CIDR (IP/tamanho do prefixo). No IPv6 o "tamanho do prefixo" é um valor decimal que especifica a quantidade de *bits* contíguos à esquerda do endereço que compreendem o prefixo. O exemplo de prefixo de sub-rede apresentado a seguir indica que dos 128 *bits* do endereço, 64 são utilizados para identificar a sub-rede (SANTOS, 2010).

- Prefixo 2001:db8:3003:2::/64

- Prefixo global 2001:db8::/32

- ID da sub-rede 3003:2

De acordo com Santos (2010), essa representação também possibilita a agregação dos endereços de forma hierárquica, identificando a topologia da rede através de parâmetros como posição geográfica, provedor de acesso, identificação da rede, divisão da sub-rede, etc. Essa representação possibilita a diminuição da tabela de roteamento e, com isso, o encaminhamento dos pacotes é agilizado.

Ainda de acordo com Santos (2010), na representação os endereços IPv6 em URLs (*Uniform Resource Locators*), passam a ser representados entre colchetes. Isso elimina a possibilidade de ambiguidades caso seja necessário indicar o número de uma porta juntamente com a URL. Veja os exemplos a seguir:

http://[2001:12ff:0:4::22]/index.html

http://[2001:12ff:0:4::22]:8080

3.13.2 Tipos de endereços IPv6

De modo geral, um endereço IPv6 faz parte de uma das seguintes categorias: *unicast, multicast* e *anycast*. Tal característica serve, basicamente, para permitir uma distribuição otimizada de endereços e possibilitar que estes sejam acessados mais rapidamente, de acordo com as circunstâncias. Veja brevemente cada um dos tipos, de acordo com Santos (2010):

- *Unicast*: este tipo de endereço identifica uma única interface, de modo que um pacote enviado a um endereço *unicast* é entregue a uma única interface.

- *Anycast*: identifica um conjunto de interfaces. Um pacote encaminhado a um endereço *anycast* é entregue à interface pertencente a este conjunto mais próximo da origem (de acordo com distância medida pelos protocolos de roteamento). Um endereço *anycast* é utilizado em comunicações de um-para-um-de-muitos.

- *Multicast*: também identifica um conjunto de interfaces; entretanto, um pacote enviado a um endereço *multicast* é entregue a todas as interfaces associadas a esse endereço. Um endereço *multicast* é utilizado em comunicações de um-para-muitos.

Diferentemente do IPv4, no IPv6 não existe endereço *broadcast* responsável por direcionar um pacote para todos os nós de um mesmo domínio. No IPv6, essa função foi atribuída a tipos específicos de endereços *multicast* (SANTOS, 2010).

Tabela 3.8: Representação dos tipos de endereços IPv6

Endereço	Representação completa	Representação abreviada
Unicast	1080:0:0:8:800:200C:417A	1080::8:800:200C:417ª
Multicast	FF01:0:0:0:0:0:0:43	FF01::43
Loopback	0:0:0:0:0:0:0:1	::1
Unspecifield	0:0:0:0:0:0:0:0	::

Fonte: http://www.rnp.br/newsgen/0103/end_ipv6.html

3.13.3 Hierarquias do endereço IPV6

Assim como acontece com o IPv4, o IPV6 também pode ter seus endereços divididos em "cotas" ou "categorias", de forma que hierarquias possam ser criadas para determinar a distribuição otimizada de endereços.

Essa hierarquia tem a alocação de todo espaço de endereçamento baseada no tipo de endereços IPv6. Nela o prefixo definido pelos primeiros *bits* do endereço IPv6 indica cada tipo de endereço. O campo variável que compreende esses *bits* é denominado *Format Prefix* (FP). A figura 3.24 mostra a hierarquização dos endereços IPv6.

Hierarquia de Endereço no IP v6 (proposta de divisão):

Prefixo Binário	Tipo de Endereço	Espaço do Endereçamento
0000 0000	Compatibilidade com IP v4	0.39%
0000 0001	Reservado	0.39%
0000 001	Endereços NSAP	0.78%
0000 010	Endereços IPX	0.78%
0000 0011	Reservado	0.78%
0000 100	Reservado	0.78%
0000 101	Reservado	0.78%
0000 110	Reservado	0.78%
0000 111	Reservado	0.78%
0001	Reservado	6.25%
001	Reservado	12.5%
010	Provedores de acesso	12.5%

011	Reservado	12.5%
100	Geográfico	12.5%
101	Reservado	12.5%
110	Reservado	12.5%
1110	Reservado	6.25%
1111 0	Reservado	3.12%
1111 10	Reservado	1.56%
1111 110	Reservado	0.78%
1111 1110	Disponível para uso local	0.39%
1111 1111	Usado para Multicast	0.39%

Figura 3.24: Proposta de a atribuição de classes para o endereço IPv6
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-redes-ip/RedesIP-Cap-02.pdf

O endereço do IPv6 atribuído para um provedor de acesso à rede (vide Figura 3.25), obedece à seguinte hierarquia:

- Cada provedor recebe uma única ID.
- O provedor atribui a cada assinante uma única ID.
- O assinante atribui uma ID a cada sub-rede e cada *host*.

Hierarquia de Endereço no IP v6 (proposta de divisão):

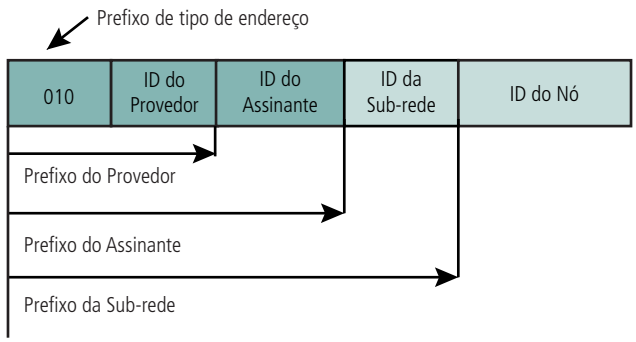


Figura 3.25: Hierarquia do endereço do IP v6 atribuído para um provedor de acesso à rede
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-redes-ip/RedesIP-Cap-02.pdf

3.13.4 IPv6 e a segurança

No IPv6, houve uma preocupação de corrigir as limitações de segurança existentes no IPv4. Por isso, segundo Kurose (2010), foram desenvolvidos novos protocolos da camada de rede que oferecem uma variedade de serviços de segurança. Um desses protocolos é o IPSec (IP *Security*), que fornece funcionalidades de criptografia de pacotes de dados, de forma a garantir três aspectos destes: integridade, confidencialidade e autenticidade.

O IPSec pode ser utilizado também no IPv4, mas possui a limitação de não poder ser usado em redes privadas baseadas em NAT. Devido à sua grande capacidade de endereçamento, o IPv6 não trabalha com o NAT, A utilização do IPSec conjuntamente com ele ocorre sem limitações.

Para efetuar sua função, o IPSec faz uso, essencialmente, de um cabeçalho de extensão chamado *Authentication Header* (AH) para fins de autenticação, de outro cabeçalho denominado *Encapsulating Security Payload* (ESP) para garantir a confidencialidade, e do protocolo *Internet Key Exchange* (IKE) para criptografia.

Vale a pena observar que o protocolo IPv6, por si só, já representa um grande avanço de segurança, uma vez que a sua quantidade de endereços é tão grande que, por exemplo, torna inviável o uso técnicas de varredura de IP em redes para encontrar possíveis computadores com vulnerabilidades de segurança. Mas, é importante frisar que, apesar do IPv6 oferecer mais segurança, isso não significa diminuir os cuidados com ela, deixando de usar outras ferramentas específicas para esse fim.

3.13.5 ICMPv6

De acordo com Gai (1998), o protocolo ICMPv6 (*Internet Control Message Protocol*) é uma parte integrante do IPv6 e combina funções anteriormente subdivididas no IPv4 pelos protocolos ICMP, IGMP e ARP.

Assim, podemos dizer que o ICMPv6 é um protocolo de uso múltiplo (por exemplo, é usado para informar erros encontrados em processamento de pacotes, executar diagnósticos, descobrir *hosts* vizinhos, etc.). Por essa razão, as mensagens de ICMP são subdivididas em duas classes: mensagens de erros e mensagens de informações. As mensagens de ICMP são transportadas dentro de um pacote IPv6.

3.14 IPv4 x IPv6

O elevadíssimo número de endereços IPv6 permite que apenas esse protocolo seja utilizado na internet. Acontece que essa mudança não pode acontecer de uma hora para outra. Isso porque roteadores, servidores, sistemas operacionais, entre outros precisam estar plenamente compatíveis com o IPv6, mas a internet ainda está baseada no IPv4. Isso significa que ambos os padrões vão coexistir por algum tempo. Mas é necessário não só que ambos coexistam, mas também se que comuniquem. Há alguns recursos criados especialmente para isso que podem ser implementados em equipamentos de rede:

- *Dual-Stack* (pilha dupla): faz com que um único dispositivo – um roteador, por exemplo – tenha suporte aos dois protocolos.

- *Tunneling* (tunelamento): cria condições para o tráfego de pacotes IPv6 em redes baseadas em IPv4 e vice-versa. Há várias técnicas disponíveis para isso, como Tunnel Broker e 6to4, por exemplo.

- *Translation* (tradução): faz com que dispositivos que suportam apenas IPv6 se comuniquem com o IPv4 e vice-versa. Também há várias técnicas para tradução, como *Application Layer Gateway* (ALG) e *Transport Relay Translator*(TRT).

Relembre alguns conceitos do IPv4, conforme Figura 3.26 a seguir.

Termo	Definição
Endereço de IP ou Endereço de *host*	Um número de 32 *bits*, normalmente escrito em formato decimal com pontos, que identifica apenas uma interface dos computadores.
Rede	Um conjunto de *host*, em que todos têm uma parte inicial idêntica nos endereços IP.
Endereço de Rede ou Número de rede	Um número de 32 *bits*, normalmente escrito em formato decimal com pontos, que representa uma rede. Esse número não pode ser atribuído como um endereço IP à interface dos computadores. A parte referente à rede do número de *host* tem um valor formado apenas por 0s binários.
Endereço de broadcast	Um número de 32 *bits*, normalmente escrito em formato decimal com pontos, usado para endereçar todos os hosts da rede. Esse número não pode ser atribuído como um endereço IP à interface dos computadores. A parte referente aos hosts tem um valor formado apenas por endereços 1s binários.
Sub-rede	Um conjunto de *host*, em que todos têm uma parte inicial idêntica nos endereços IP. Uma sub-rede difere de uma rede à medida que ela é uma subdivisão de uma rede, com uma parte maior dos endereços sendo idêntica.
Endereço de sub-rede ou Número de sub-rede	Um número de 32 *bits*, normalmente escrito em formato decimal com pontos, que representa uma sub-rede. Esse número não pode ser atribuído como um endereço IP à interface dos computadores. A parte referente aos *hosts* tem um valor formado apenas por 0s binários.
Sub-redes	O resultado da subdivisão das redes em sub-redes menores. Esse é o jargão, por exemplo, "Você está criando sub-redes?"
Máscara de rede	Um número de 32 *bits*, normalmente escrito em formato decimal com pontos. A máscara é usada pelos computadores para calcular o número de rede de um determinado endereço IP fazendo um AND Booleano no endereço IP e na máscara. A máscara também define o número de *bits* de *host* em um endereço.
Máscara	Um termo genérico para máscara, quer seja uma máscara-padrão, quer seja uma máscara de sub-rede.

Máscara padrão Classe A	A máscara usada em redes Classe A quando as sub-redes não estão sendo usadas. O valor é 255.0.0.0.
Máscara padrão Classe B	A máscara usada em redes Classe B quando as sub-redes não estão sendo usadas. O valor é 255.255.0.0.
Máscara padrão Classe C	A máscara usada em redes Classe C quando as sub-redes não estão sendo usadas. O valor é 255.255.255.0.
Parte ou campo de rede	Termo usado para descrever a primeira parte da um endereço IP. A parte que justamente representa a rede. Está parte depende da máscara escolhida.
Parte ou campo de host	Termo usado para descrever a última parte de um endereço IP. Está parte depende da máscara escolhida.

Figura 3.26: Conceitos do IPv4
Fonte: http://proginf7.blogspot.com.br/2010/09/enderecamento-ip.html

Assista ao filme *Guerreiros da Net*, disponível em http://pontasdamadrugada.blogspot.com/2007/03/warriors-of-net-guerreiros-da-internet.html, e poste suas observações sobre o funcionamento do protocolo IP no AVEA.

Resumo

Nesta aula você conheceu o endereço IPV4, viu como ele é constituído e como funciona. Conheceu as classes e subclasses do endereço IPV4 e também os endereços especiais e privados. Viu também que ele necessita da máscara de sub-redes como parte de seu endereçamento. Conheceu alguns serviços que atuam com ele, como o DHCP e o NAT. Viu as suas novas formas de endereçamento CIDR e VLSM. Percebeu que o endereço IPv4 está se exaurindo e, por causa disso, foi desenvolvido o IPV6. Viu como ele funciona e como é usado. Por último, pôde fazer uma comparação entre o IPV4 e o IPV6.

Leia a reportagem no jornal Estado de Minas sobre esgotamento de endereços IPv4, disponível em http://www.em.com.br/app/noticia/tecnologia/2011/06/02/interna_tecnologia,231546/para-evitar-o-ipcalipse.shtml

Para aprender mais sobre o IPv6, existe um curso on-line muito bom em: http://curso.ipv6.br/

Veja um estudo comparativo IPv4 X IPv6 em http://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2009_2/priscilla/ipv6_cabecalho.html

Atividades de aprendizagem

Responda às questões a seguir e poste no AVEA da disciplina.

1. Se uma sub-rede tem endereço de rede como 200.201.5.32 com máscara 255.255.255.224, qual o último endereço válido para um equipamento nessa sub-rede?

2. Uma empresa precisa dividir uma classe C em 32 sub-redes. Quantos *bits* de rede deverão ser definidos em 1 na máscara de sub-rede?

3. Em uma rede classe A, qual será o endereço de máscara de rede quando cinco *bits* forem utilizados para compô-la?

4. Um administrador de rede IP decidiu utilizar uma máscara 255.255.248.0. Faça o raciocínio inverso do utilizado pelo administrador para justificar essa escolha.

Assista ao vídeo "Do IPv4 até o IPv6", disponível em http://olhardigital.uol.com.br/produtos/central_de_videos/do_IPv4_para_o_ipv6_voce_sabe_o_que_significa_esta_mudanca, e poste seus comentários no ambiente um texto com suas impressões sobre essa mudança

Dado o endereço IP 200.200.200.10 com a máscara 255.255.248.0 responda:

5. Qual o endereço de rede do respectivo IP?

6. Qual o endereço IP utilizado para *broadcast*?

7. Considere a situação onde uma empresa necessite de 520 endereços IP para a interligação de duas redes. Qual seria a máscara oferecida a essa empresa a fim de otimizar a utilização de endereços IPs? Qual seria o endereço CIDR correspondente?

8. Considerando o endereço IP 150.60.10.1, identifique:

A parte ID de rede: _____

A parte ID de *host*: _____

Classe: _____

Aula 4 – Equipamentos básicos de conectividade

Objetivos

Conhecer roteamento, seus algoritmos e protocolos.

Conhecer protocolos da camada de rede.

Saber sobre os elementos ativos de uma rede de computadores.

Vários são os aspectos que devem ser estudados e pesquisados com o objetivo de se alcançar a melhor lógica de interconexão em uma rede de computadores. É necessário valorizar a *performance*, o gerenciamento, a forma de interligação, o custo x benefício e os meios físicos utilizados para o tráfego da informação. Dessa maneira, os elementos de uma rede que interagem produzindo sistemas de comunicação balanceados e estruturalmente equilibrados podem ser definidos como elementos fundamentais no processo de comunicação em rede LAN, MAN ou WAN.

Segundo Lages (2012), uma rede não é só feita de estações, servidores e cabos. Existem dispositivos que podem ser usados para expandir a rede, segmentar o tráfego e para conectar duas ou mais redes. Esses dispositivos são chamados de equipamentos de conectividade.

Dentre esses equipamentos podem-se destacar os repetidores e os *hubs* que funcionam como repetidores para toda a rede do sinal recebido de um *host* dessa rede, as pontes e *switches* que têm como principal função a segmentação do tráfego em uma rede com vários *hosts*, os roteadores que servem como meio de interligação de duas ou mais redes e os *gateways* que tornam possível a comunicação entre diferentes ambientes e arquiteturas. Ainda é necessário mencionar as placas de redes, cuja importância é inegável, apesar de não serem equipamentos de conectividade, pois são a interface dos *hosts* com o meio físico.

Tabela 4.1: Equipamento de conectividade X camada de atuação

Equipamento de conectividade	Camada RM-OSI	N. Camada
Gateway de aplicação	Camada de aplicação	7
Gateway de transporte	Camada de transporte	4
Roteador	Camada de rede	3
Ponte, *switch*	Camada de enlace	2
Repetidor, *hub*	Camada física	1

Fonte: Elaborada pelo autor

Em suas funções, os equipamentos básicos de conectividade em uma rede devem proporcionar:

- velocidade e *performance* de transmissão de informação;
- aprimoramento ou modelagem da informação;
- gerenciamento seguro da informação e processos (qualitativo e quantitativo);
- abrangência do espectro da comunicação (maior número de usuários);
- facilidade na interoperabilidade dos equipamentos pragmaticamente projetados para trabalharem em conjunto;
- segurança de destinação e conteúdo;
- velocidade de transmissão;
- abrangência de distâncias limites;
- distribuição de dados inteligentemente direcionados;
- flexibilização nas conversões entre plataformas diferenciadas.

4.1 Placas de rede

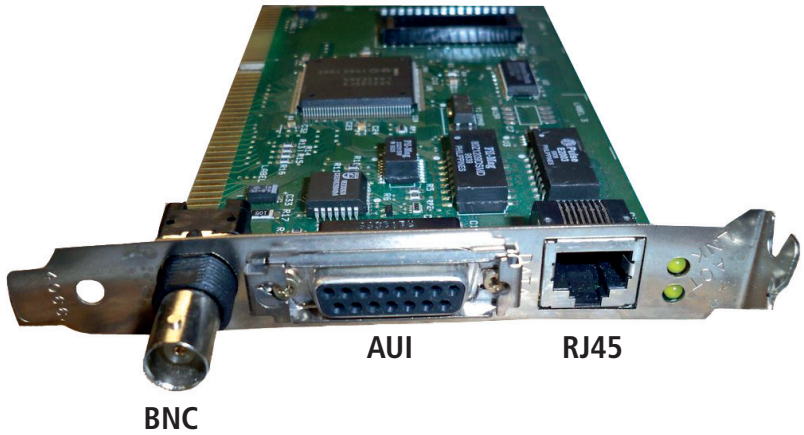


Figura 4.1: Exemplo de uma placa de rede
Fonte: http://lpozza.blogspot.com.br/2008_12_01_archive.html

Para que um *host* (computador) seja ligado a uma rede, é necessário que ele possua uma interface que o adéque ao tipo de rede a que ele estará ligado. Esse "interfaceamento" é função da placa de rede.

Placas de rede ou NICs (*Network Interface Cards*) como são popularmente conhecidas, atuam como a interface física entre os computadores e o cabo da rede. São instaladas nos *slots* de expansão de cada computador ou servidor. Após a NIC ter sido instalada, o cabo da rede é ligado a uma de suas portas. Ela tem as seguintes funções (LAGES, 2012):

- Preparar dados do computador para o cabo da rede.
- Enviar os dados para outro computador.
- Controlar o fluxo de dados entre o computador e o sistema de cabeamento.
- Receber os dados vindos do cabo e traduzi-los em *bytes* para ser entendido pelo computador.

Uma placa de rede é definida de acordo com o padrão de rede ao qual o *host* estará ligado, sendo os padrões mais comuns: Ethernet, ATM, *token ring*, FDDI e FO. É necessário também que as placas possuam tipos de conectores diferentes de acordo com o meio físico em que elas estão inseridas. Os conectores mais comuns são os RJ45, BNC e AUIX (Figura 4.1). Atualmente o conector mais usado é o RJ45 com o padrão Ethernet. São comuns também as placas *wireless* que funcionam sem fio, cujo padrão mais usado é também o Ethernet.

Como todo *hardware* integrante de um computador, as placas de rede também precisam ser configuradas. Atualmente essas configurações são feitas por *hardware* (normalmente *straps*), *software*, através do método *plug and play*. Algumas placas possibilitam BOOT remoto, onde não é necessário que o computador precise de sistema operacional instalado, o que pode ser muito útil em sistemas cliente/servidor, pelo qual o servidor pode inicializar o cliente, através de perfis predefinidos, que ele carrega diretamente do servidor. Nesse caso o computador funciona como um terminal, sem a capacidade de armazenamento. Apesar de as placas de rede não serem um equipamento de conectividade, é explícita sua importância nas redes computacionais.

4.2 Repetidores

Os repetidores são os equipamentos de conectividade mais simples que existem. São utilizados, geralmente, para a interligação de duas ou mais redes idênticas. Atuando no nível físico, eles simplesmente recebem todos os pacotes de cada uma das redes que interligam e os repetem nas demais redes sem realizar qualquer tipo de tratamento sobre eles (SOARES, 1995).

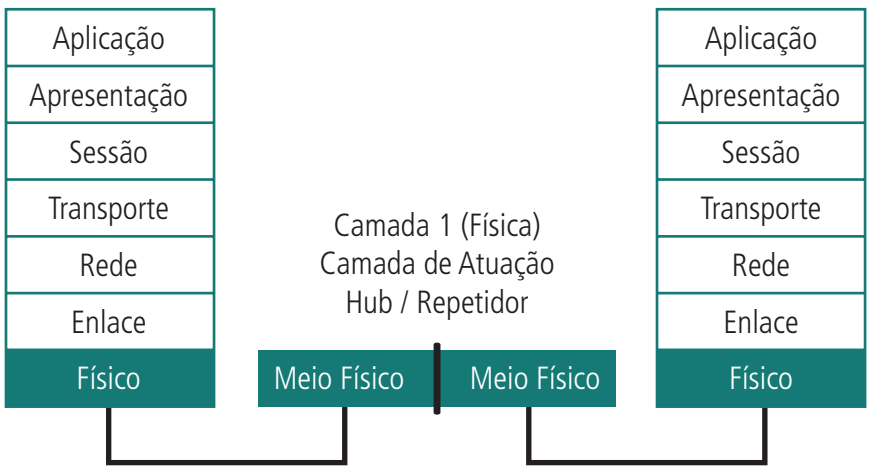


Figura 4.2: Rede com repetidor
Fonte: Elaborada pelo autor

Como os repetidores recebem dados do barramento de uma determinada rede e envia-os (repete-os) para todos os outros barramentos a ele interligados, inclusive para o barramento de onde ele recebeu os dados, as redes que eles interligam se comportam como uma única rede (um único domínio de colisão), possuindo inclusive o mesmo endereço lógico. Mas ao repetir todas as mensagens que recebe, um tráfego extra e inútil é gerado pelo repetidor quando os pacotes repetidos não se destinam às redes que interligam.

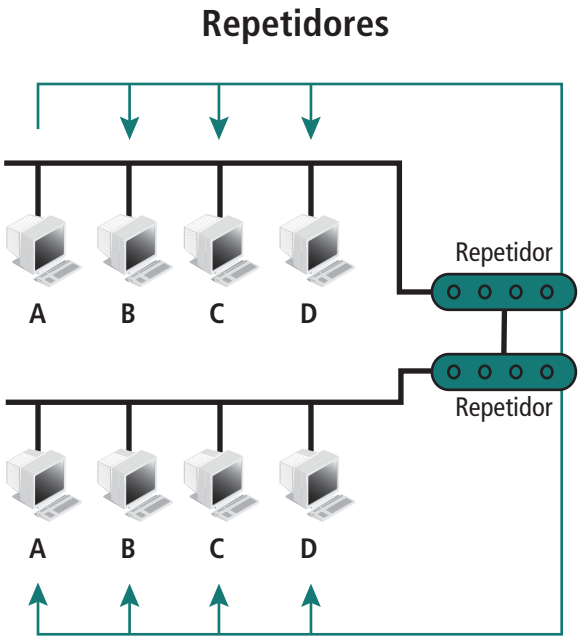


Figura 4.3: Exemplo de rede com repetidores
Fonte: http://www.dee.ufcg.edu.br/~scaico/facisa/irc/6a_-_repetidores,_hubs,_pontes_e_switches_%28slides%29.pdf

Os repetidores, como as placas de redes, têm seus padrões definidos de acordo com a rede a qual estão inseridos. Assim, eles podem ser repetidores Ethernet, repetidores ATM, repetidores *token ring* (que também são chamados de MAU), repetidores FDDI, etc. E fisicamente, os repetidores podem ser construídos com duas (*dualport*) ou mais (*multiport*) portas.

De acordo com Torres (2009), ao interligar várias redes padrão Ethernet, através de vários repetidores, é necessário obedecer à regra de segmentação, na qual a comunicação entre a primeira rede e a última não pode passar por mais de quatro repetidores.

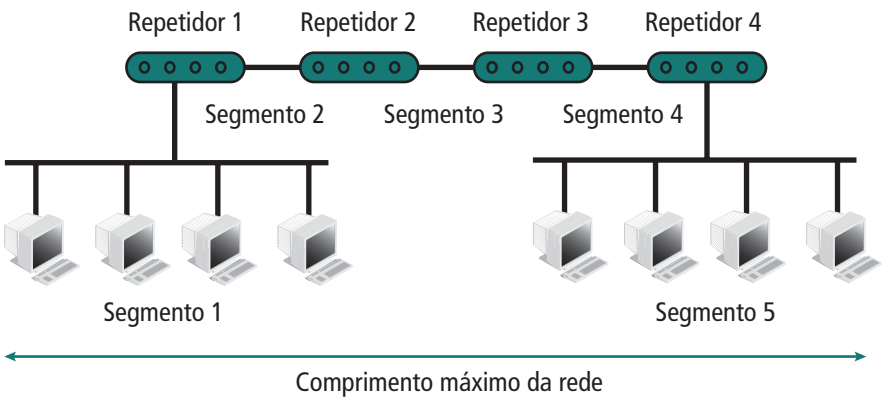


Figura 4.4: Padrão de segmentação dos repetidores
Fonte: http://www.dee.ufcg.edu.br/~scaico/facisa/irc/6a_-_repetidores,_hubs,_pontes_e_switches_%28slides%29.pdf

Pontos a serem considerados em redes com repetidores:

- Em redes de anel onde a estação é responsável pela retirada dos próprios quadros, cabe ao repetidor retirar os quadros onde ele atua como re-transmissor.

- Em redes onde o destinatário retira o pacote, o repetidor não pode fazê-lo, diminuindo o desempenho da rede.

- Nas redes de protocolos baseados por contenção, cabe aos repetidores gerenciar as colisões.

- Em uma rede com vários repetidores não pode haver um caminho fechado entre dois repetidores, pois isso implicaria duplicações infinitas de quadros.

- Os pacotes são repetidos para todas as redes, quando seria necessário apenas transmiti-los para a rede de destino.

- Em protocolos nos quais o reconhecimento de quadros é realizado nos próprios quadros transmitidos, essa característica é perdida, uma vez que não pode ser realizada pelos repetidores. Primeiro, pela possibilidade de existirem vários repetidores na rede, para qual deles caberá a tarefa? E também se esse problema fosse contornado, como o repetidor escolhido poderia saber da situação, no quadro na situação de destino, uma vez que ainda nem o transmitiu?

4.3 *Hubs*

De acordo com Lages (2012), *hubs* são exemplos de repetidores multiportas (*multiport*), e como tal atuam na camada 1 (física) do modelo RM-OSI. Nesse esquema de atuação, quando ele recebe um sinal em uma porta, ele regenera e amplifica esse sinal, retransmitindo-o para todas as suas portas, incluindo aquela em que ele recebeu o sinal.

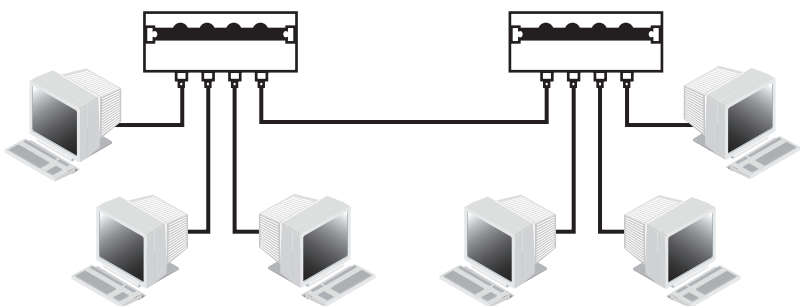


Figura 4.5: Ligação em rede com *hub*
Fonte: http://www.bertozi.com/adb/PITAGORAS/Sistemas/Gerencia_de_Redes/apostila%20REDES%20TOTAL.pdf

Todo *hub* é um repetidor (mas nem todo repetidor é um *hub*). Ele é responsável por replicar em todas as suas portas as informações recebidas pelas máquinas da rede. Por exemplo, se uma máquina quer enviar um quadro de dados para a outra, todas as demais máquinas da rede recebem esse quadro de dados ao mesmo tempo (TORRES, 2009).

Por serem multiportas, as conexões da rede com os *hubs* são concentradas (por isso ele é chamado de concentrador). E como concentradores eles se configuram, fisicamente, em redes com topologia em estrela, facilitando o gerenciamento da rede e a solução de problemas, pois os *hosts* (computadores) são conectados às portas do *hub* e se houver algum problema em um *host*, a rede não será afetada, somente aquela porta. A rede só será paralisada se o *hub* apresentar algum problema.

Segundo Torres (2009), se fisicamente (relativo ao *layout* físico utilizado na instalação da rede), os *hubs* estão ligados na topologia em estrela, logicamente eles representam uma topologia em barra (vide Figura 4.6) ou anel, com vários *hubs* interconectados. A topologia lógica é aquela observada sob o ponto de vista das interfaces das estações com a rede e com os *hubs*. Essa particularidade se deve ao fato de os *hubs*, assim como os concentradores, trabalharem na camada 1 (física) do modelo RM-OSI. Essa característica faz com que eles recebam os dados enviados por um *host* ligado em uma de suas portas, regenerem-nos, amplifiquem-nos e retransmita-os para todas as suas portas, inclusive para a que gerou o sinal. Devido a essa característica, diz-se que os *hubs* não segmentam o tráfego, estando no mesmo domínio de colisão. Os *hubs*, também como os concentradores, são definidos de acordo com o tipo de rede a qual estão inseridos: Ethernet, *token ring* (MAU), ATM, FDDI, etc.

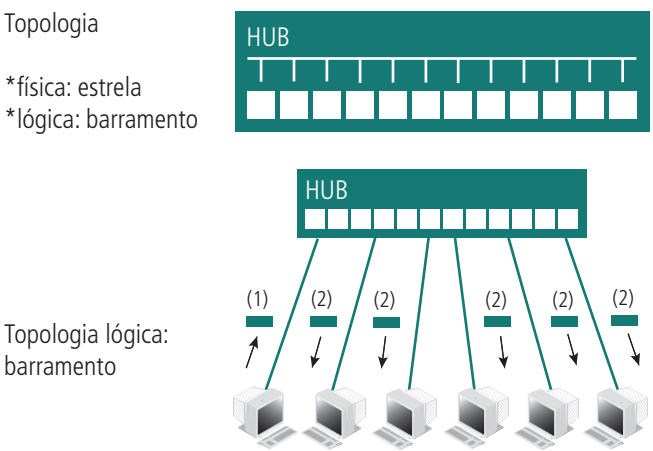


Figura 4.6: Rede com *hub*
Fonte: www.lsi.usp.br/~volnys/courses/tecredes/pdf/03FISICO-col.pdf

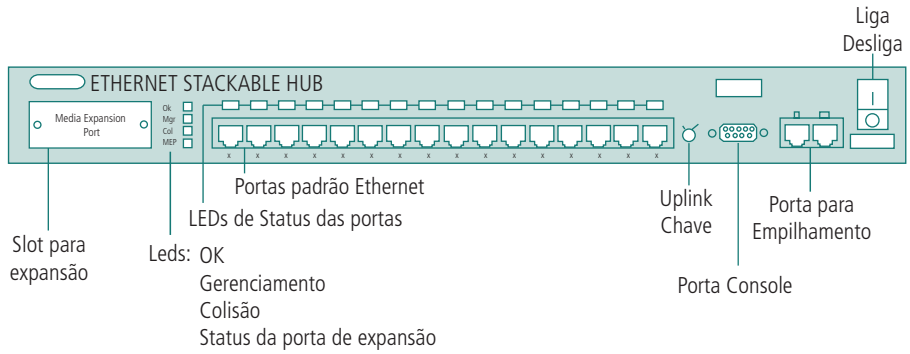


Figura 4.7: *Hub* padrão Ethernet IBM 8224
Fonte: http://publibfp.dhe.ibm.com/cgi-bin/bookmgr/BOOKS/e2ei8000/1.1?DT=19941006170102#FIGBOXDIAG

Os *hubs* podem estar ligados entre si, aumentando assim o número de portas disponíveis. Essa conexão entre *hubs* pode ser feita através da técnica de cascateamento e empilhamento.

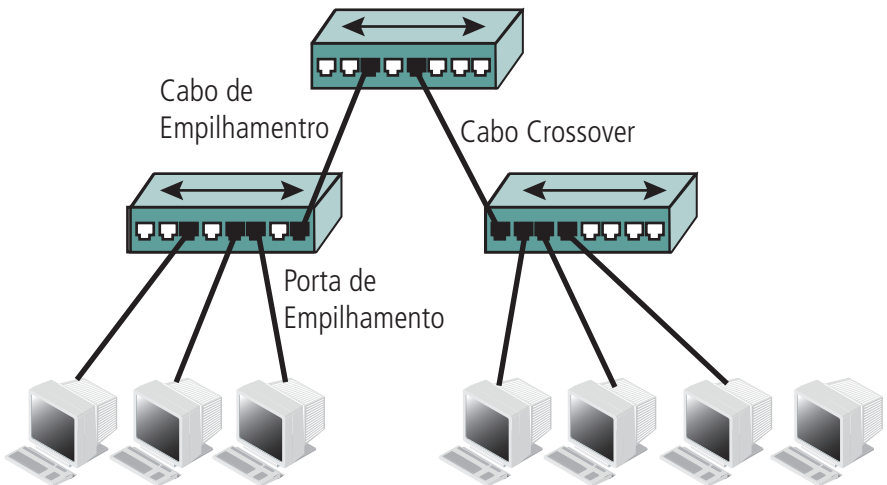


Figura 4.8: Rede com *hubs*
Fonte: http://pt.kioskea.net/contents/lan/concentrateurs.php3

4.3.1 Cascateados (ligados porta a porta)

Os *hubs*, como qualquer elemento de conectividade, possuem o limite físico de número de portas, normalmente o máximo de 48 portas. Assim, à medida que a rede cresce, podem-se conectar *hubs* para atender a essa demanda. Uma delas é ligá-los em série através de cabos par trançado pelas suas portas normais, que é o método chamado de cascateamento.

Nesse tipo de ligação, os *hubs* são ligados de tal maneira que formam um único conjunto. Assim, qualquer sinal transmitido por um *host* interligado ao conjunto será retransmitido para todos os outros, pois os *hosts* ficam no mesmo domínio de colisão. E quanto mais *hosts* estiverem conectados,

maior será o tráfego e mais lenta ficará a rede, pois apesar de existirem limites para conexão entre *hubs*, o único limite para o número de portas que um *hub* pode ter é apenas físico.

Outro inconveniente nesse tipo de ligação é que entre o equipamento que inicia a transmissão e o que recebe não pode ter mais de quatro passos (ou quatro ligações) e cada vez que se passa por um *hub* tem-se um passo. Então nesse percurso só podem existir no máximo três *hubs* cascateados para interligá-los.

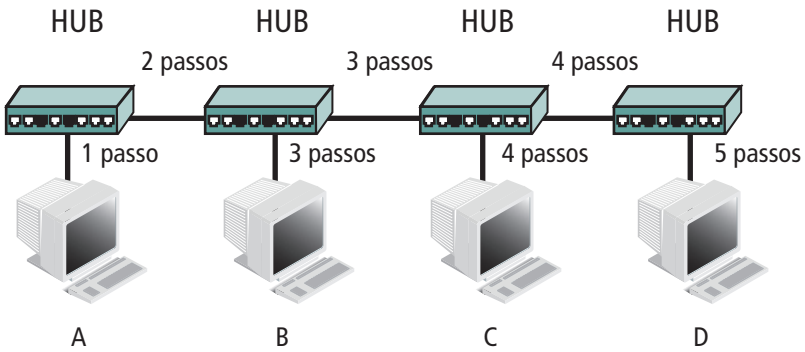


Figura 4.9: Problema do cascateamento de *hub*
Fonte: Elaborada pelo autor

No padrão Ethernet, para interligar os *hosts* ao *hub*, o cabo usado é o *straight-through*, que é padrão dos cabos RJ45 (cabo RJ45 normal). Mas para cascatear *hubs* é necessário o uso de cabos RJ 45 *crossover*. A regra básica, no padrão Ethernet é que quando os equipamentos ligados forem iguais, ou estiverem na mesma camada (física, enlace), o cabo é *crossover;* e quando os equipamentos forem de camadas diferentes, o cabo usado é *straight-through*.

Padrão dos cabos RJ45:

- **Cabo *straight-through***: 1 – Verde/branco; 2 – Verde; 3 – Laranja/branco; 4 – Azul; 5 – Azul/branco; 6 – Laranja; 7 – Marrom/ branco; 8 – Marrom.

- **Cabo *crossover***: 1 – Laranja/branco; 2 – Laranja; 3 – Verde/branco; 4 – Azul; 5 – Azul/branco; 6 – Verde; 7 – Marrom/branco; 8 – Marrom.

- **Cabo *crossover***: uma ponta *crossover*, outra ponta *straight-through.*

- **Cabo normal**: duas pontas iguais *straight-through*.

Alguns *hubs* Ethernet permitem cascatear *hubs* por uma porta especial chamada de *uplink*, que é uma porta que tem os pinos invertidos, o que permite a interligação de *hubs* com o cabo *straight-through.*

4.3.2 Empilhados (*stackables*)

Para minimizar os problemas descritos no cascateamento, foram desenvolvidos *hubs* que possibilitam uma ligação especial chamada *stackable*, também conhecida como empilhamento. Essa ligação é feita por uma porta especial, permite a conexão entre dois ou mais *hubs*, o que faz com que estes sejam considerados pela rede como um só *hub* e não *hubs* separados. O inconveniente nesse processo é que ele só funciona com *hubs* da mesma marca. Uma vantagem do empilhamento é que quatro *hubs* empilhados contam de 1 a 1,5 passos, possibilitando o empilhamento de até oito *hubs*.

A interligação através dessa porta específica com um cabo específico de empilhamento tem velocidade de transmissão maior que a velocidade das portas. Normalmente o *hub* é composto de duas portas para empilhar, pois uma recebe o sinal e a outra o envia.

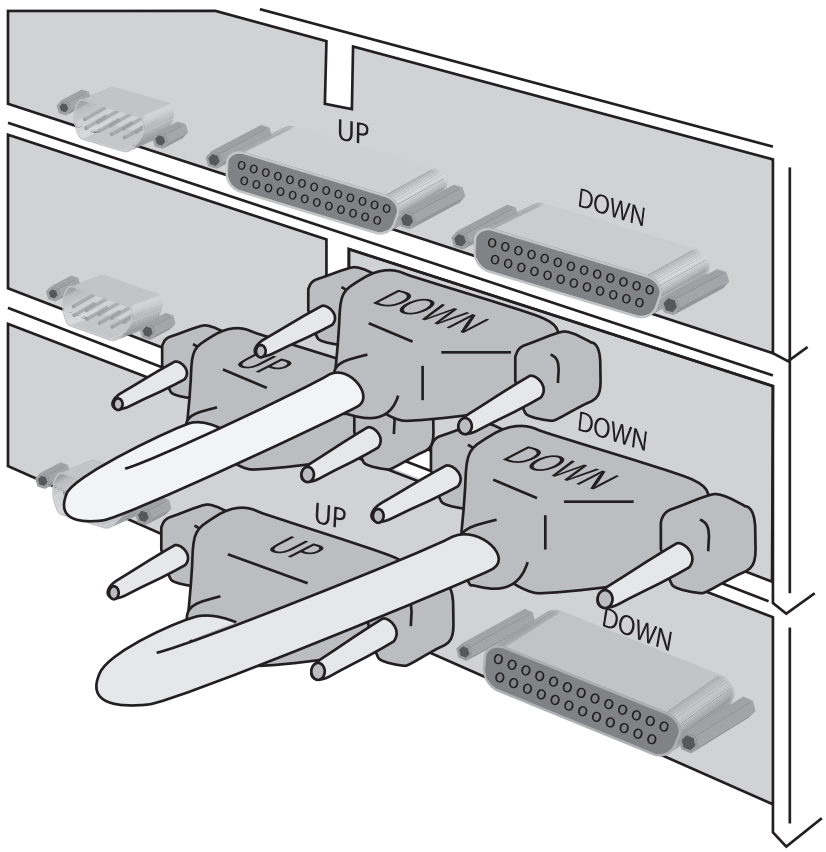


Figura 4.10: Empilhando *hubs* pela porta especial
Fonte: http://pt.scribd.com/doc/30413789/Apostila-Redes-Com-Put-Adores-Lages

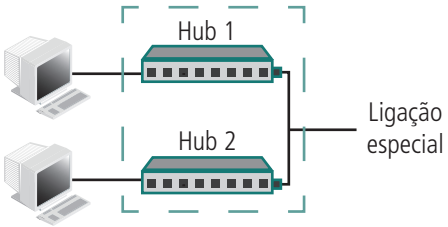


Figura 4.11: *Hubs* empilhados
Fonte: http://www.dee.ufcg.edu.br/~scaico/facisa/irc/6a_-_repetidores,_hubs,_pontes_e_switches_%28slides%29.pdf

4.3.3 *Hubs* inteligentes

Segundo Lages (2012), alguns *hubs*, apesar de trabalharem na camada física do modelo OSI, podem ter características de gerenciamento e são chamados de *hubs* gerenciáveis ou inteligentes. Isso é bastante útil para o caso de empilhamento de *hubs* (vários *hubs* interligados). Eles vão além das funções desempenhadas pelos *hubs* comuns, pois possibilitam:

- incorporar um processador e *softwares* de diagnóstico;
- detectar e mesmo isolar da rede estações problemáticas;
- detectar pontos de congestionamento;
- interagir através de uma interface de linha de comando;
- impedir acesso não autorizado ao equipamento;
- gerenciar através de *software* possibilitado por um módulo que pode ser acoplado ao *hub*.

Normalmente os *hubs* inteligentes utilizam o protocolo SNMP para fazer o gerenciamento.

No protocolo SNMP há dois elementos distintos: o gerente, ou agente, e o objeto gerenciado. Há uma troca contínua de informações entre eles para montar um estado da rede. Essas informações são armazenadas na MIB (*Management Information Base*) que fica no agente.

Embora o *hub* inteligente aparente a ideia de que ele consegue filtrar ou isolar o tráfego entre os *hosts*, isso não é feito por ele, pois, assim como os repetidores, ele continua atuando na camada 1 do modelo RM-OSI. E quando um *host* ligado a uma pilha de *hubs* transmite, o seu sinal é propagado

por todas as portas do *hub* e empilhamento e, consequentemente, para toda a rede, ou seja, os *hosts* fazem parte do mesmo domínio de colisão, e por isso compartilham o mesmo segmento e a mesma largura de banda. O motivo se deve ao fato de que, atuando na camada 1 do modelo RM-OSI, ele não consiga trabalhar com nenhum tipo de endereçamento. À medida que uma rede composta por *hubs* cresce muito e os problemas de *performance* passam a aparecer, são necessários outros tipos de equipamentos de conectividade que podem ajudar a segmentar o tráfego e aumentar a *performance* da rede.

4.4 Pontes (*Bridges*)

A ponte opera na camada 2 (enlace) do modelo OSI, ou seja, ela é capaz de entender endereços MAC e portanto de filtrar tráfego entre segmentos de uma rede. Como a ponte opera na camada 2, ela permite que qualquer tipo de protocolo passe por ela. Ela é muito útil quando precisamos segmentar uma rede grande em duas redes menores para aumentar a *performance* (LAGES, 2012).

As pontes atuam como filtros ao segmentar redes. Assim elas repassam todos os quadros que são destinados a nodos que não pertençam ao mesmo segmento dos nodos de origem, isolando o tráfego interno dos segmentos para as outras porções da rede, melhorando o tempo de resposta ao usuário.

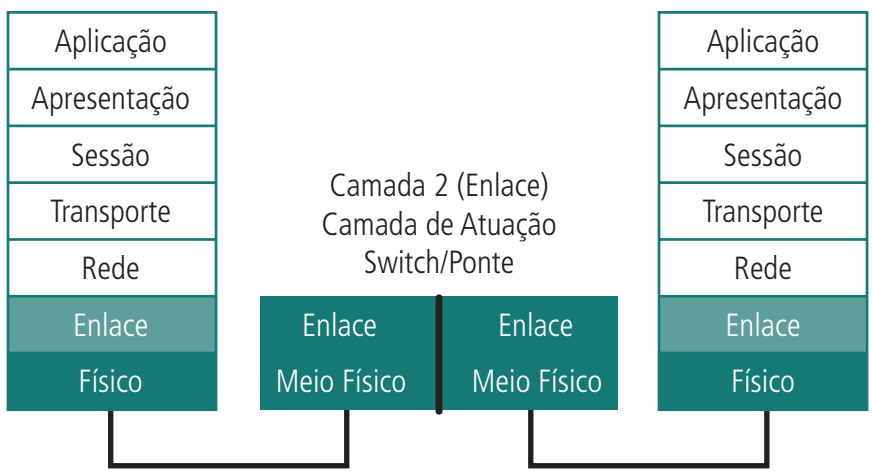


Figura 4.12: Pontes
Fonte: Elaborada pelo autor

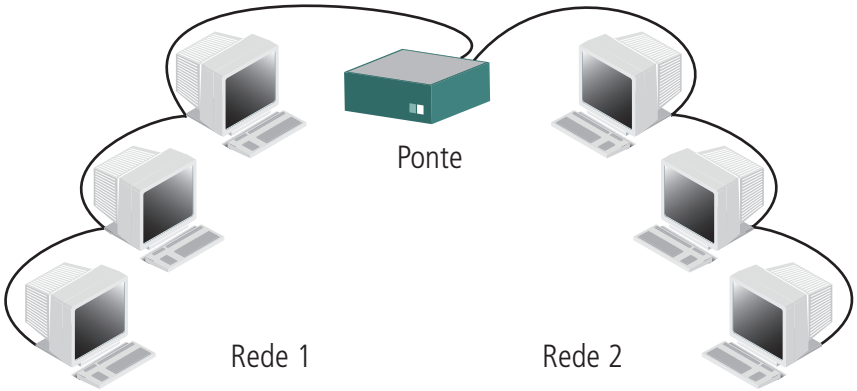


Figura 4.13: Funcionamento das pontes
Fonte: http://pt.scribd.com/doc/30413789/Apostila-Redes-Com-Put-Adores-Lages

A Figura 4.12 mostra duas redes interligadas por uma ponte. Nesse esquema de conexão, quando um *host* da rede 1 quiser se comunicar com outro *host* que está na mesma rede, o tráfego gerado não atravessa para a rede 2 (Figura 4.13). Porém, quando um *host* da rede 1 quer se comunicar com um *host* na rede 2, a ponte permite que o tráfego chegue à rede 2 (Figura 4.14). Essa filtragem é feita baseada no endereço MAC dos *hosts*.

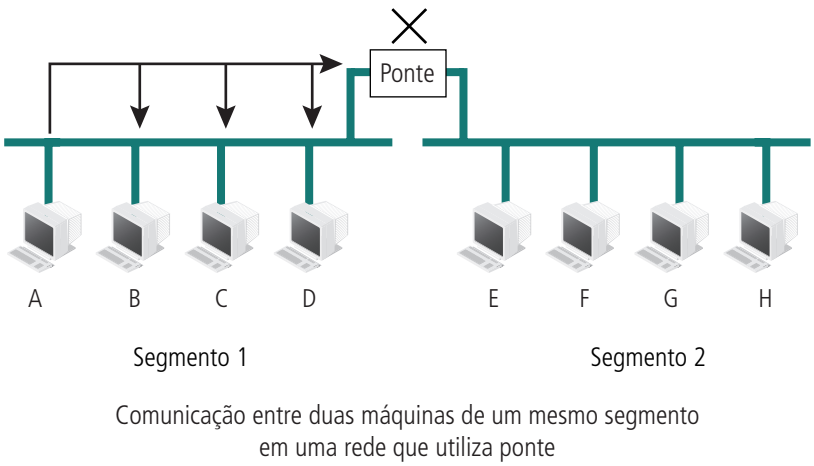
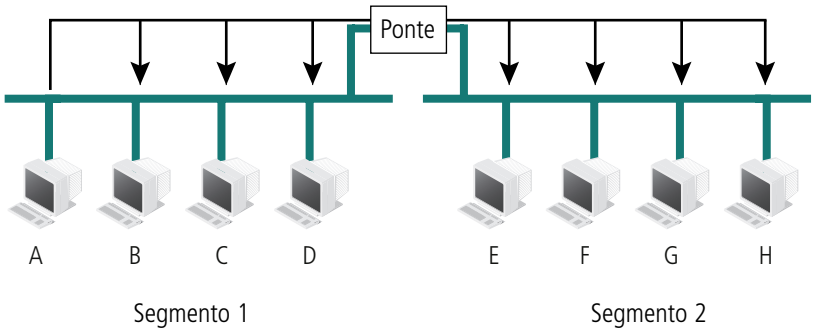


Figura 4.14: Ponte filtrando o tráfego
Fonte: http://www.dee.ufcg.edu.br/~scaico/facisa/irc/6a_-_repetidores,_hubs,_pontes_e_switches_%28slides%29.pdf



Comunicação entre duas máquinas de segmentos distintos em uma rede que utiliza ponte

Figura 4.15: Ponte deixando o tráfego passar
Fonte: http://www.dee.ufcg.edu.br/~scaico/facisa/irc/6a_-_repetidores,_hubs,_pontes_e_switches_%28slides%29.pdf

Operando baseada no princípio de que cada máquina tem o seu endereço MAC, a ponte constrói uma tabela relacionando o endereço MAC à rede a qual ele pertence, e baseada nessa tabela é que ela toma as decisões na filtragem do tráfego. Mas, como ao ser ligada à ponte não conhece nenhum endereço MAC, essa tabela é construída com base na verificação dos endereços MAC origem de cada quadro. Isso é feito quando um *host* envia um quadro por uma rede interligada à ponte. Ela então analisa esse quadro e associa o endereço MAC dele com sua rede de origem. Essa tabela é volátil, ou seja, quando a ponte é desligada, essas informações são perdidas.

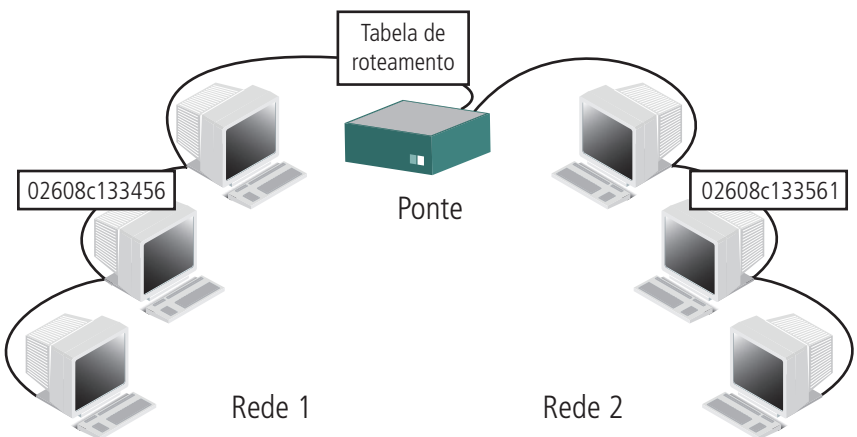


Figura 4.16: Ponte construída de endereços MAC
Fonte: http://www.dee.ufcg.edu.br/~scaico/facisa/irc/6a_-_repetidores,_hubs,_pontes_e_switches_%28slides%29.pdf

Como as pontes manipulam quadros, elas não retransmitem ruídos, erros, ou quadros de formação ruim (um *frame* deve estar completamente válido para ser transmitido por uma ponte), ao contrário dos repetidores e *hubs*, que, por atuarem na camada 1 do modelo RM-OSI, manipulam sinais elétricos, podendo transmitir um quadro de formação ruim.

São atribuições das pontes:

- filtrar as mensagens de tal forma que somente aquelas endereçadas a ela sejam tratadas;
- armazenar mensagens quando o tráfego for muito grande;
- ler o endereço do quadro e retransmiti-lo apenas para a porta de destino;
- filtrar as mensagens, de modo que quadros com erros não sejam retransmitidos;
- funcionar como uma estação repetidora comum;
- poder atuar como elementos passivos gerenciadores de rede, e pode coletar dados estatísticos de tráfego de pacotes para elaboração de relatórios.

Segundo Soares(1995), existem pontes locais e remotas. As pontes locais oferecem uma conexão direta entre múltiplos segmentos de LANs numa mesma área, enquanto que as remotas conectam múltiplos segmentos de redes locais em áreas dispersas. Existem também pontes que oferecem as duas funções, sendo porém menos frequentes.

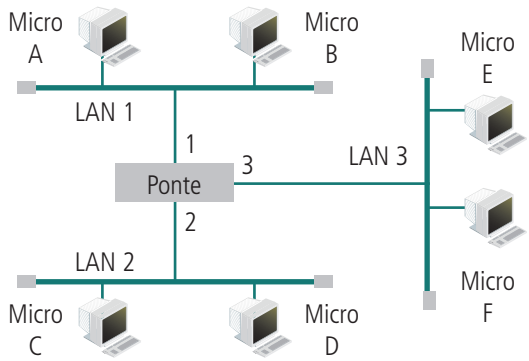
Como já dito, as pontes podem ser utilizadas para conectar redes similares (Ethernet com Ethernet, *token ring* com *token ring*) ou redes diferentes (Ethernet com *token ring*, Ethernet com FDDI). De acordo com Soares (1995), quando se interligam redes similares às pontes, geralmente utiliza-se o mecanismo das pontes transparentes ou o mecanismo das pontes com roteamento na origem.

4.4.1 Pontes transparentes

As redes locais interligadas por pontes transparentes não sofrem nenhuma modificação ao serem interconectadas por esses equipamentos, que são transparentes para os nodos da rede. Ao serem ligadas, as pontes passam a analisar o endereço de origem dos quadros enviados por todos os segmentos ligados a ela, e concluem que o nodo de origem pode ser atingido através da porta pela qual o quadro chegou. Com esse mecanismo, as pontes constroem uma tabela de rotas, que é composta por pares que contêm o endereço de origem e a porta de saída associada a ele. Porta é a denominação que se dá a cada ligação da ponte a uma LAN, cada uma com um endereço MAC diferente.

As pontes transparentes operam abaixo da interface definida pelo serviço MAC. A denominação transparente deve-se ao fato de as LANs não sofrerem nenhuma modificação ao serem interconectadas por esse tipo de ponte (SOARES, 1995).

Ao receber um quadro, a ponte verifica na tabela de rotas se o endereço de destino dele está associado a uma porta diferente da de origem, enviando, nesse caso, o quadro para a porta indicada. Por outro lado, se a tabela não possuir nenhuma associação ao endereço de destino, o quadro é retransmitido para todas as portas (*flooding*), exceto para a porta de origem. Pacotes de *broadcast* e *multicast* são também enviados dessa forma.



Micro	Porta

Ponte é ligada
Tabela Vazia

Micro	Porta
A	1

Tabela da ponte após uma transmissão de A para D

Micro	Porta
A	1
E	3

Tabela da ponte após uma transmissão de E para A

Micro	Porta
A	1
E	3
B	1

Tabela da ponte após uma transmissão de B para C

Figura 4.17: Aprendizado da ponte
Fonte: Elaborada pelo autor

O esquema de aprendizado de endereços das pontes transparentes funciona bem quando não existem rotas alternativas na inter-rede. A existência de rotas alternativas implica na formação de ciclos (caminhos fechados) na topologia da rede, o que provoca falhas no funcionamento da estratégia anteriormente descrita.

Um exemplo de cenário onde o mecanismo de aprendizado de rotas das pontes transparentes não funciona corretamente se apresenta da seguinte forma: suponhamos que duas redes, A e B, sejam interligadas por duas pontes, P1 e P2 (a duplicação das pontes aumenta a confiabilidade da inter-rede); suponhamos ainda que a estação X esteja ligada à rede A, e a estação Y à rede B. Inicialmente a tabela de rotas das pontes está vazia.

Nesse momento a estação X envia um quadro para Y. As pontes P1 e P2 capturam o quadro através da rede A com endereço de origem X e adicionam uma entrada em suas tabelas de rotas, associando o endereço X à rede A. Uma vez que as pontes não sabem em que rede fica a estação de destino do quadro, no caso Y, ambas retransmitem o quadro para a rede B. Assim, Y recebe duas cópias do mesmo quadro. Essa não é, entretanto, a pior falha do método, pois o fato de a ponte P1 receber o quadro retransmitido por P2, e vice-versa, faz com que elas atualizem incorretamente suas tabelas de rotas. E mais, enquanto durar esse estado inconsistente, as pontes permanecem retransmitindo quadros duplicados.

Veja o que acontece quando a ponte P2 recebe o quadro retransmitido por P1. O endereço de origem desse quadro é X, e dessa vez ele é recebido por P2 na rede B. Logo, a ponte P2 conclui equivocadamente que a estação X está ligada à rede B, e atualiza a entrada de sua tabela de rotas relativa à estação X, associando-a a rede B. Enquanto essa informação for mantida na tabela de rotas, os quadros endereçados a X não serão roteados corretamente. Uma vez que P2 continua não sabendo onde fica a estação Y, que é o destino do quadro, ela retransmite o quadro na rede A. A ponte P1, ao receber novamente o quadro pela rede A, retransmite-o para a rede B, pois também não sabe ainda em que rede está ligada a estação Y. Esse procedimento fica então se repetindo enquanto Y não transmitir um quadro. O problema aqui exemplificado é contornado pelo padrão IEEE 802.1D.

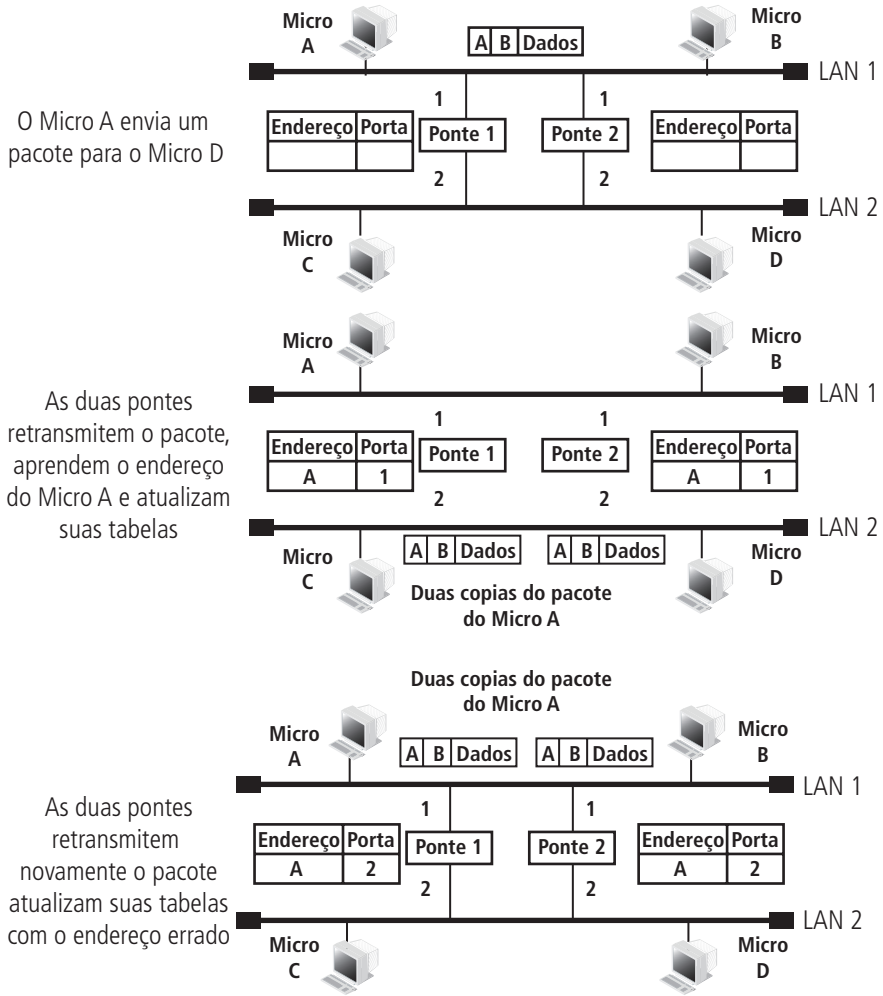


Figura 4.18: Problema do *loop*
Fonte: Elaborada pelo autor

4.4.2 Pontes com roteamento na origem

As pontes transparentes têm a vantagem de serem de fácil instalação, mas não fazem o melhor uso da banda passante, uma vez que utilizam apenas um subconjunto da topologia, a árvore geradora. Esses e outros fatores levaram à escolha do esquema chamado de ponte com roteamento na origem.

Nesse esquema, a estação de origem escolhe o caminho que o quadro deve seguir e inclui a informação de roteamento no cabeçalho do quadro. A informação de roteamento é construída da seguinte forma: cada LAN possui um identificador único, e cada ponte possui um identificador único no contexto das redes às quais está conectada. Uma rota é uma sequência de pares (identificador de rede, identificador de ponte).

O primeiro *bit* do endereço de origem dos quadros cujo destino não está na mesma rede da estação de origem é igual a 1.

Ao escutar um quadro cujo primeiro *bit* do endereço de origem é igual a 1, a ponte analisa a informação de roteamento do quadro. Se o identificador da LAN através da qual ele chegou é seguido pelo identificador da ponte em questão, ela retransmite o quadro na rede cujo endereço vem depois do seu identificador na informação de roteamento do quadro. Quando o número da LAN de onde veio o quadro não é seguido pelo seu identificador, a ponte não retransmite o quadro.

Tabela 4.2: Comparação entre pontes transparentes e de roteamento pela origem

Tema	Ponte transparente	Ponte de roteamento pela origem
Orientação	Sem conexões	Baseada em conexão
Transparência	Totalmente transparente	Não transparente
Configuração	Automática	Manual
Roteamento	Abaixo de ótimo	Ótimo
Localização	Aprendizado às avessas	Quadros de descoberta
Falhas	Tratadas pelas pontes	Tratadas pelos *hosts*
Complexidade	Nas pontes	Nos *hosts*

Fonte: Elaborada pelo autor

4.5 *Switches*

Segundo Torres (2009), os *switches* são pontes com várias portas e, como tal, atuam até a camada 2 (enlace) do modelo RM-OSI. Essa característica permite a eles segmentarem o tráfego, pois em vez de replicar os dados recebidos para todas as suas portas, os *switches* enviam os quadros somente para o *host* que os requisitou, trazendo uma melhora considerável no desempenho da rede.

Ao enviar o quadro somente para o *host* requisitante, o *switch* faz uma comutação virtual entre os *hosts* de origem e destino, isolando as demais portas desse processo. Essa característica permite que a comunicação ocorra em modo *full-duplex*, diferentemente de como acontece com repetidores, *hubs* e ponte que compartilham a banda. Essa operação de comutação, segundo Lages (2012), é chamada de *switching*.

O *switch* provê uma filtragem de pacotes entre LANs que estejam separadas, mas, ao contrário da ponte, que usa um barramento interno compartilhado, ele permite várias comunicações *host* a *host* simultaneamente, já que comuta quadros utilizando caminhos dedicados. Essa comutação é feita através da subcamada MAC, da camada de enlace, que possui o endereço físico da placa de rede do *host*.

Quando uma máquina envia um quadro para a rede através do *switch*, este lê o campo de endereço MAC de origem do quadro e anota em uma tabela interna o endereço MAC da placa de rede do micro que está conectada àquela porta (TORRES, 2009).

Como o *switch* segmenta o tráfego entre suas portas, colisões não ocorrerão; porém, poderá ser experimentada a contenção de dois ou mais quadros que necessitem do mesmo caminho ao mesmo tempo. Esses quadros são armazenados em *buffers* de entrada e saída das portas e são transmitidos posteriormente assim que os caminhos estiverem desocupados, graças aos *buffers* de entrada e saída das portas.

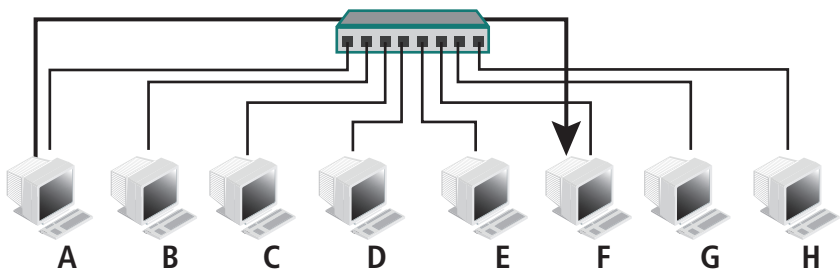


Figura 4.19: Rede com *switch*
Fonte: http://willamys.files.wordpress.com/2011/05/aula3-equipamentos-de-redes-pontes-e-switches.pptx

De acordo com Lages (2012), é a tabela CAM, no *switch*, que relaciona as portas com os endereços MAC dos *hosts*. Quando o *switch* precisa encaminhar um quadro e não há em sua tabela qualquer informação sobre em qual porta está o *host* de destino, ele encaminha o quadro para todas as portas, exceto para a porta que originou o *frame*. Depois que o *host* responde, ele armazena a relação endereço MAC/porta na tabela CAM, daí em diante ele passa a se comunicar diretamente com o *host* através daquela porta. O aprendizado e atualização da tabela são realizados por um processador central no *switch*, que pode também proporcionar tarefas de gerenciamento, como atualizar a MIB SNMP e manter tabelas de redes locais virtuais. Outra situação em que o quadro é encaminhado a todas as portas do *switch* é quando o quadro é um *broadcast*; isso é, o endereço MAC e destino é FFFF. Nesses dois tipos de situação, o *switch*, obviamente, não segmenta tráfego.

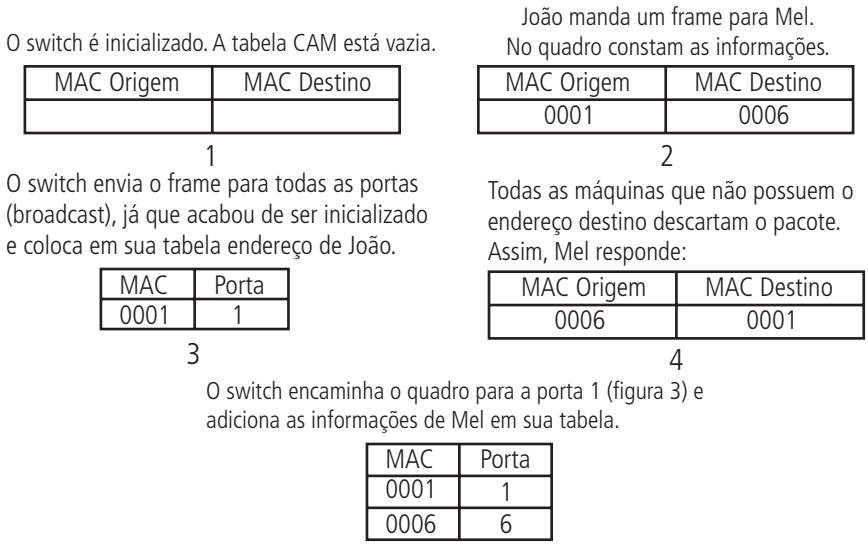


Figura 4.20: Construção da tabela CAM
Fonte: http://pt.scribd.com/doc/30413789/Apostila-Redes-Com-Put-Adores-Lages

4.5.1 Modo de operação do *switch*

Segundo Pinheiro (1995), os *switches* trabalham principalmente em três modos de operação: *store-and-forward*, *cut-through* e *fragment-free*.

No esquema *store-and-forward* o quadro deve ser recebido completamente antes de ser iniciada a transmissão para o endereço destino. O quadro recebido é armazenado no *buffer* da porta de entrada ou saída, dependendo da arquitetura. Depois de ele estar todo no *buffer* é que ele é transmitido para o destinatário. Uma vantagem desse modo é que, uma vez que os quadros foram recebidos inteiros, é possível realizar um controle de erros e descartar os pacotes com problemas, o que não é possível no modo *cut-through*, que transmite os quadros sem verificar erros.

No esquema *cut-through* os quadros são enviados assim que chegam ao *buffer* da porta de entrada ou saída, dependendo da arquitetura. Quando o quadro chega ao *switch*, seu endereço destino é comparado na tabela a fim de verificar a porta de saída. Desde que esta porta esteja disponível (não esteja sendo usada no momento para nenhuma outra transmissão), o quadro começa a ser imediatamente enviado. Essa transmissão ocorre em paralelo com o recebimento do restante do quadro pela porta de entrada. O *switch* no modo *cut-through* reverte para o modo *store-and-forward* quando a porta destino está ocupada, revertendo novamente para o modo *cut-through*, quando a porta já estiver disponível.

O esquema *fragment-free* trabalha muito similarmente ao *cut-through*, exceto pela particularidade de que os primeiros 64 *bytes* do pacote são armazenados antes que ele seja transmitido. A razão disso é que a maioria dos erros e todas as colisões ocorrem nos primeiros 64 *bytes* do pacote. Esse é o modo menos comum de operação dos *switches*.

Tabela 4.3: Comparação entre os modos de operação *switch*

	Cut-through/Fragment-free	Store-and-forward
Latência	Menor	Maior
Dependência quanto aos tamanhos dos pacotes	Menos	Mais
Descarte de quadros com erros	Não há descarte	Há descarte

Fonte: Elaborada pelo autor

Ainda existe uma classificação secundária que divide os *switches* em: *switch* de *workgroup*, que suporta somente uma estação ligada por porta; e *switch* de *backbone* congestionado, que suporta segmentos com múltiplas estações em cada porta.

4.5.2 Tipos de *switch*

Os *switches* possuem vários tipos de projetos físicos diferentes, sendo os mais comuns: *Shared-memory*, *Matrix* e *Bus-architeture*.

- ***Shared-memory***: armazena todos os pacotes de entrada em um *buffer* comum a todas as portas; então, manda os pacotes para as portas corretas relacionadas ao nó de destino.

- ***Matrix:*** este tipo de *switch* possui uma rede interna com as portas de entrada e de saída cruzadas umas com as outras; quando um pacote é detectado numa porta de entrada, o endereço MAC é comparado com a tabela de portas de saída; encontrada a porta, o *switch* faz a conexão entre os dois nós.

- ***Bus-architeture:*** em vez de uma grade, ele possui um caminho interno de transmissão dividido para todas as portas usando TDMA; nesta configuração cada porta tem um *buffer* de memória dedicado.

Como já foi dito, os *switches*, normalmente, atuam na camada 2 (Enlace) do Modelo RM-OSI, mas já existe os *switches* chamados de *core* que trabalham na camada 2 e 3 do modelo RM-OSI, incorporando características dos roteadores e passando a trabalhar na camada de rede. Esses *switches* mais

potentes possuem microprocessadores internos que garantem a eles um poder de processamento capaz de traçar os melhores caminhos para o tráfego dos quadros, evitando a colisão dos pacotes e ainda conseguindo tornar a rede mais confiável e estável.

Devido às suas melhores características e preço, os *switches* estão se tornando os equipamentos concentradores mais usados em redes locais (LAN).

4.6 VLAN

Segundo Morales (2012), VLAN (*Virtual Local Area Network* ou *Virtual LAN* ou Rede Local Virtual), é a divisão de uma rede local física em rede local logicamente independente. Numa rede local, a comunicação entre as diferentes máquinas é governada pela arquitetura física. Nas VLANs é possível definir uma segmentação lógica (*software*) baseada num agrupamento de máquinas segundo alguns critérios de elemento ativo (endereços MAC, números de porta, protocolo, etc.). O principal elemento ativo a usar VLANs são os *switches*, com os quais as VLANs podem coexistir, de maneira a formar várias redes virtuais para criar domínios de *broadcast* separados. Uma VLAN também permite colocar em um mesmo domínio de *broadcast*, *hosts* com localizações físicas distintas e ligadas a *switches* diferentes.

Um *switch* com uma VLAN implementada tem múltiplos domínios *broadcast* e funciona de maneira semelhante a um roteador (TYSON, 2012).

As redes virtuais, por trabalharem em *switches*, operam na camada 2 do modelo RM-OSI (com endereço MAC); no entanto, elas são, geralmente, configuradas para mapear diretamente uma rede ou sub-rede IP.

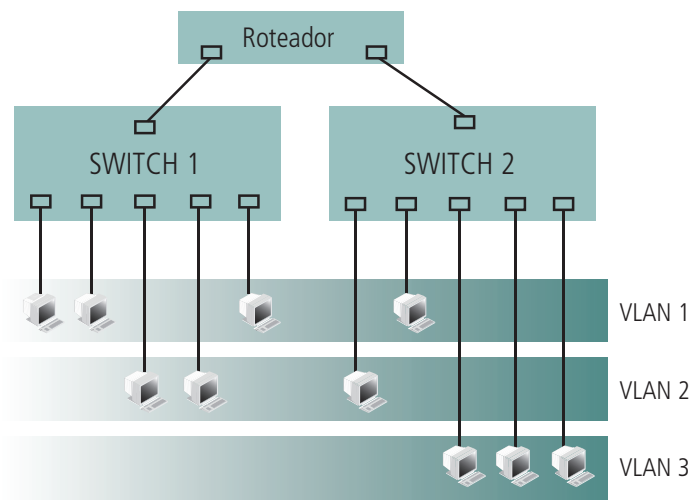


Figura 4.21: *Switches* com VLANs
Fonte: Elaborada pelo autor

Saiba como funcionam os *switches*, acessando http://www.malima.com.br/switchs-como-funcionam/ e http://informatica.hsw.uol.com.br/lan-switch.htm

Assista à apresentação de "Por dentro das coisas – *switch*" disponível em http://www.youtube.com/watch?v=QU03X09EPgs&feature=related

Mídias integradas: Acesse a animação de "Como o *switch* LAN trabalha", disponível em http://www.cisco.com/image/gif/paws/10607/lan-switch-transparent.swf. Teste todas as opções no menu à esquerda e poste no ambiente um texto sobre a relação nessa animação com o que foi estudado sobre *switches* neste aula.

4.6.1 Tipos de VLANs

De acordo com Morales (2012), os tipos de VLANs são definidos de acordo com o critério de comutação e o nível em que atuam:

- uma VLAN de nível 1 (também chamada VLAN por porta, em inglês *port--based* VLAN) define uma rede virtual em função das portas de conexão;

- uma VLAN de nível 2 (igualmente chamada VLAN MAC, em inglês MAC a*ddress-based* VLAN) consiste em definir uma rede virtual em função dos endereços MAC das estações. Este tipo de VLAN é muito mais flexível que a VLAN por porta, porque a rede é independente da localização da estação;

- uma VLAN de nível 3, também chamada de VLAN por sub-rede (em inglês *Network Address-Based* VLAN) associa sub-redes de acordo com o endereço IP fonte dos datagramas. Esse tipo de solução confere uma grande flexibilidade, na medida em que a configuração dos comutadores se altera automaticamente no caso de deslocação de uma estação. Por outro lado, uma ligeira degradação de desempenhos pode fazer-se sentir, dado que as informações contidas nos pacotes devem ser analisadas mais detalhadamente.

4.6.2 Vantagens no uso de VLANs

A VLAN permite definir uma nova rede lógica, ou virtual, acima da rede física, cujas vantagens são as seguintes:

- mais flexibilidade para a administração e as modificações da rede porque qualquer arquitetura pode ser alterada por simples parametrização dos comutadores;

- ganho em segurança, porque as informações são encapsuladas num nível suplementar e são eventualmente analisadas;

- redução da divulgação do tráfego sobre a rede.

4.7 *Gateways*

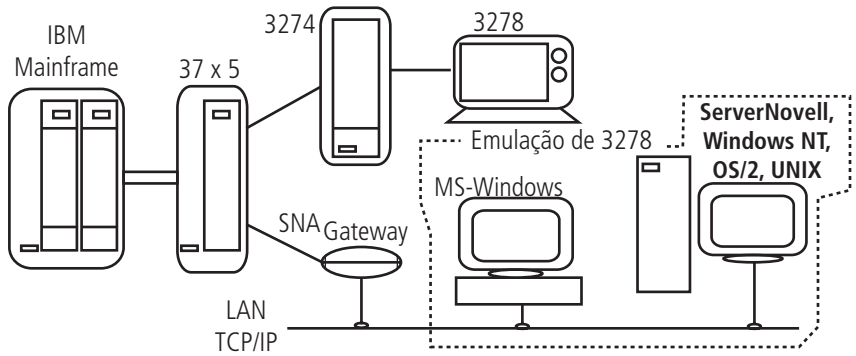


Figura 4.22: Modelo de *gateway*
Fonte: http://www.ic.uff.br/~ferraz/REDES/RCCAP09.DOC

Segundo Lages (2012), o *gateway* tem como função fazer a interligação de redes distintas (usando protocolos distintos, com características distintas). Ele atua em todas as camadas do modelo RM-OSI, resolvendo problemas de diferença entre as redes que interliga, tais como: tamanho dos pacotes que transitam nas redes, forma de endereçamento, temporizações, forma de acesso, padrões de linguagem interna de formato de correios eletrônicos. A Figura 4.22 mostra um exemplo de um *gateway* interligando várias redes distintas.

Um *gateway* liga dois sistemas que não usam os mesmos protocolos de comunicação, a mesma estrutura de formatação de dados, a mesma linguagem, a mesma arquitetura (LAGES, 2012).

De acordo com Soares (1995), os *gateways* são usualmente classificados em conversores de meio (*media-conversion gateway*) e tradutores de protocolos (*protocol-translation gateway*).

4.7.1 *Gateways* tradutores de protocolos

Os gateways tradutores de protocolo trabalham na camada 7 (aplicação) do modelo RM-OSI e, segundo Soares (1995), são mais utilizados em inter-redes que utilizam circuitos virtuais passo a passo. Eles atuam traduzindo mensagens de uma rede para outra, com a mesma semântica de protocolo. Por exemplo, o open em uma rede poderia ser traduzido por um call request em outra rede ao passar pelo gateway.

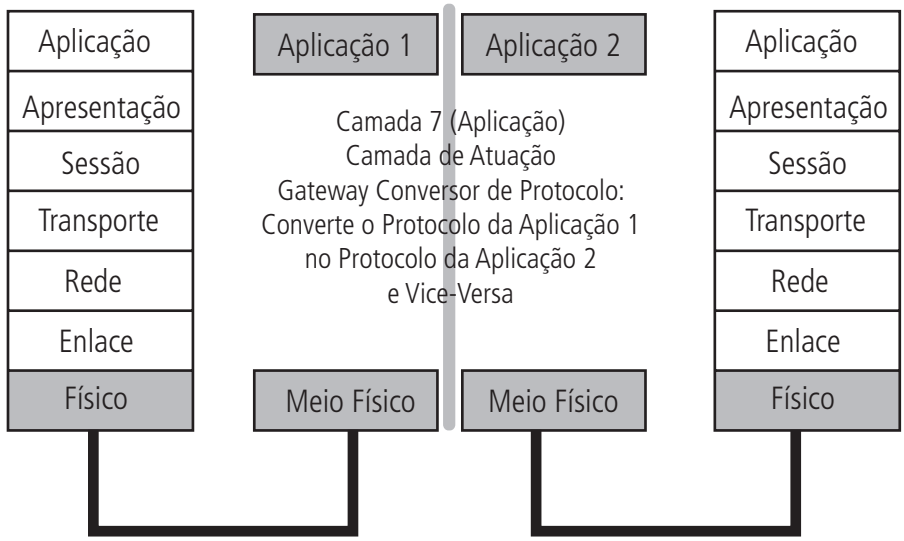


Figura 4.23: *Gateway* tradutor de protocolo
Fonte: Elaborada pelo autor

É importante observar que nem todos os protocolos podem ser mapeados entre si, e que o subconjunto formado pela interseção dos serviços comuns é o serviço que deverá ser oferecido como base para a interligação. As traduções dos protocolos são bastantes complexas e de difícil realização, o que pode aumentar em muito o custo da interligação.

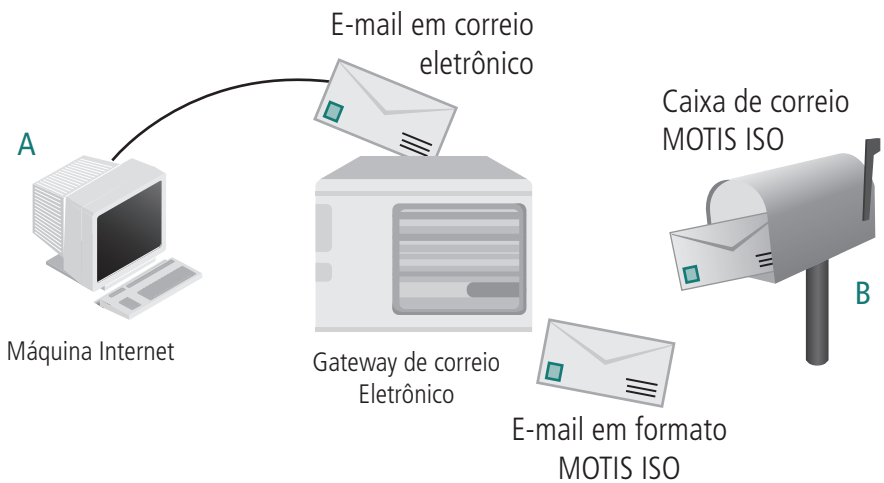


Figura 4.24 *Gateways* de aplicação
Fonte: https://disciplinas.dcc.ufba.br/pastas/MAT055/UFBA-interconexao_de_redes.ppt

4.7.2 *Gateways* conversores de meio

De acordo com Soares (1995), os *gateways* conversores de meio são mais simples por trabalharem na camada 3 (rede) do modelo RM-OSI e são bastante utilizados em inter-redes que oferecem o serviço de datagrama. Sua função é receber um pacote da camada 4 (transporte), tratar o cabeçalho inter-redes do pacote, descobrindo o caminho necessário para este, construir o novo pacote com novo cabeçalho inter-redes e enviar esse novo pacote ao próximo destino, segundo o protocolo da rede local em que este se encontra. A Figura 4.25 mostra o nível do RM-OSI onde atua o *gateway* conversor de meio.

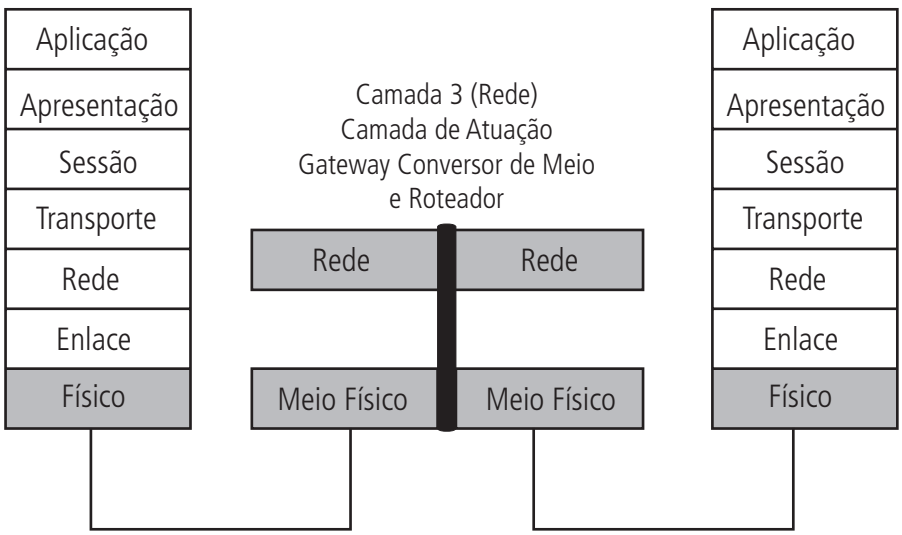


Figura 4.25: *Gateway* conversor de meio
Fonte: https://disciplinas.dcc.ufba.br/pastas/MAT055/UFBA-interconexao_de_redes.ppt

Para saber mais sobre o funcionamento de um *gateway*, acesse http://www.rederio.br/ceo/introducao/gateways.html

4.8 Roteadores

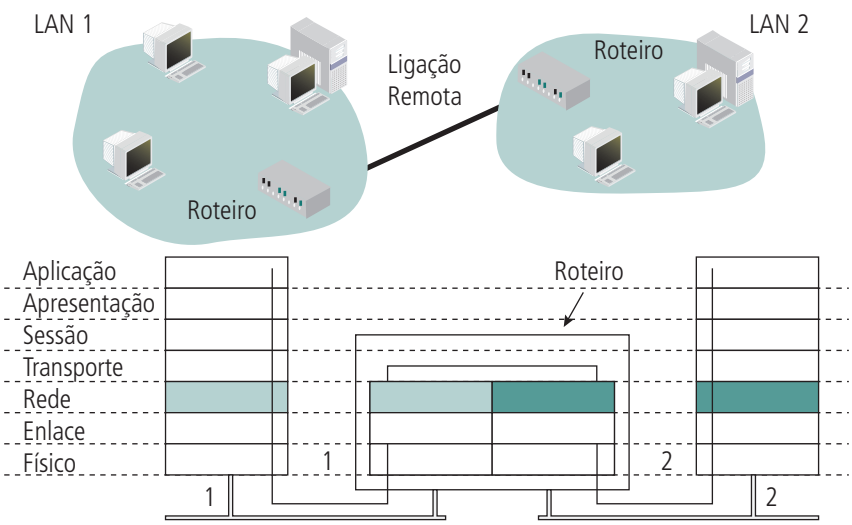


Figura 4.26: Nível do RM-OSI onde atuam os roteadores
Fonte: http://www.ic.uff.br/~ferraz/REDES/RCCAP08.DOC

Roteador é um nome mais específico para *gateway* conversor de meio. O roteador é usado em redes que possuem segmentos. Ele é configurado para saber o endereço de cada segmento. De acordo com Lages (2012), o roteador tem a capacidade de determinar qual o melhor caminho ou melhor rota para envio de dados, além de filtrar o tráfego de *broadcast*. Essa capacidade de determinar o melhor caminho inclui uma série de regras de roteamento, como as rotas estáticas inseridas no roteador, e as rotas dinâmicas aprendidas através de protocolos de roteamento (RIP, OSPF, etc.). Essa função de roteamento mostra que a área de atuação de um roteador vai até a camada 3 (rede) do modelo RM-OSI.

Os roteadores leem as informações de endereçamento de rede contidas nos pacotes e estabelecem uma conexão entre as redes. Eles têm a responsabilidade de saber como toda a rede está conectada, para poder transmitir os pacotes dos remetentes até os destinatários, passando de uma parte da rede para outra. Em poucas palavras, eles evitam que os *hosts* percam tempo assimilando conhecimentos sobre a rede.

"O papel fundamental do roteador é poder escolher um caminho para o datagrama chegar ate o seu destino. Em redes grandes pode haver mais de um caminho, e o roteador é o elemento responsável por tomar a decisão de qual caminho percorrer." (TORRES, 2009).

Os roteadores oferecem muito mais flexibilidade no tráfego de uma rede, podendo interligar as várias as redes divididas em grupos lógicos ou físicos. Para tanto, é necessário que o protocolo com o qual a rede trabalhe possibilite o roteamento, como por exemplo, o TCP/IP.

4.8.1 Funcionamento dos roteadores

Segundo Torres (2009), por trabalharem na camada 3 do modelo RM-OSI, os roteadores filtram o seu tráfego baseados nos campos de endereçamento contidos dentro do cabeçalho do protocolo de rede, que no caso do TCP/IP é o endereço IP. Subcampos desses endereços identificam o segmento da LAN onde estão localizados os *hosts* de origem e de destino. Protocolos em que os campos de endereçamento não possuem subcampos identificadores da localização do *host* de destino não podem ser roteados, podendo, porém, ter seus pacotes filtrados por *switches*.

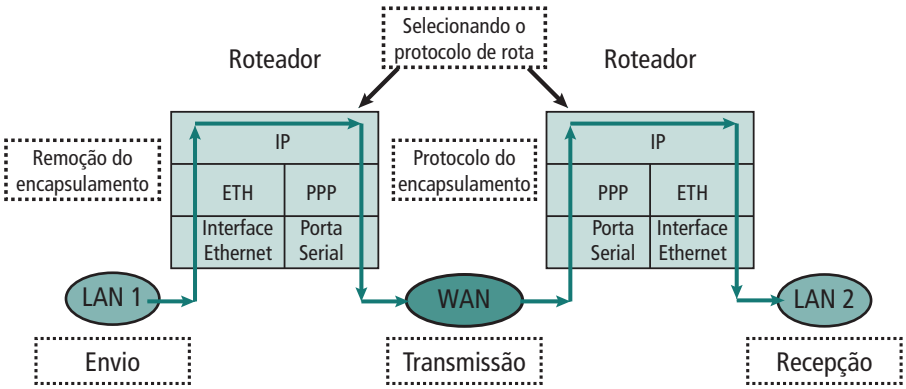


Figura 4.27: Roteadores definem as rotas
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-redes-ip/RedesIP-Cap-01.pdf

Os roteadores só entendem os endereços que pertencem aos segmentos de rede ligados a eles, portanto eles não falam com computadores remotos. Quando um roteador recebe um pacote cujo destino é uma rede remota, ele encaminha esse pacote para o outro roteador conectado a ele, e esse outro roteador fará a mesma coisa até que esse pacote chegue à rede destino. Segundo Torres (2009), a cada roteador pelo qual o pacote tem que passar dá-se o nome de *hops* ou pulo.

Observe pela figura 4.28 que o pacote para ir do remetente até o destinatário tem que passar por três roteadores e, portanto, o número de *hops* ou pulos necessários para fazer a rota é três.

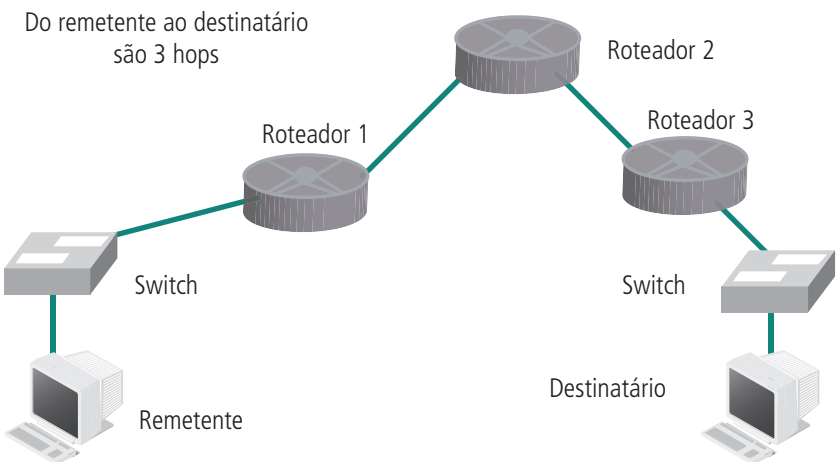


Figura 4.28: Exemplo de *hops*
Fonte: http://pt.scribd.com/doc/30413789/Apostila-Redes-Com-Put-Adores-Lages

4.8.2 Roteadores internos e externos

Segundo Comer (2007), existem dois tipos de roteamento: aquele que é encaminhado através de uma única rede, ou roteamento interno, e o encaminhamento indireto, ou roteamento externo.

Os roteadores internos são aqueles formados pela combinação de placa de comunicação síncrona/assíncrona, PC, placa Ethernet e SW de roteamento carregado ou *bult-in* no S.O do PC. Trata-se de uma solução fácil e de baixo custo. Porém, existe dependência com as características técnicas do servidor em que reside: tipo de *bus* e sistema operacional. Se um desses dois componentes muda (digamos que você muda o servidor de Intel para PA RISC), então todo investimento do roteador interno (HW + SW) é perdido.

Os roteadores externos são aqueles formados por HW e SW dedicados ao roteamento. Por ter funções exclusivamente voltadas ao roteamento, sua *performance* atinge índices superiores, justificando até um custo ligeiramente maior que a dos roteadores internos. O produto nesse caso é independente da arquitetura de HW/SW do roteador, pois, tipicamente, estamos falando de ligação ao roteador via Ethernet e *host* do TCP/IP, ou outro protocolo de comunicação.

Fazem parte dos roteadores externos os seguintes componentes de HW com função relacionada:

- Módulo LAN: para conexão com rede local.
- Módulo WAN: para conexão remota com outras redes locais.
- Módulo de processamento: CPU de alta capacidade de processamento.
- Módulo RAM: para armazenar dados voláteis, dados dinamicamente alterados.
- Módulo *flash*: para armazenar *firmware* (Kernel e SW de roteamento).
- Módulo CMOS: para armazenar dados não voláteis (configuração, por exemplo).
- Módulo console: para gerenciamento e configuração do roteador.
- Módulo de temporização: para temporizações do SW de roteamento.

4.8.3 Aplicações de roteadores

Os roteadores são bem utilizados no meio internet/intranet e para comunicação LAN-to-LAN como, por exemplo, ligação entre a matriz e a filial de uma organização. No meio internet/intranet, o roteador aparece na ligação do *site* do provedor (rede local do provedor) ao *link* internet, bem como na conexão do provedor a subprovedores via linha de dados especializada ou não especializada. A ligação matriz–filial pode ser feita pela internet para esse fim, apenas ressaltando que deve ser usado algum artifício de proteção nas pontas para evitar acesso público, já que a internet é aberta. Um exemplo dessa proteção é a **VPN**.

A-Z

VPN
(*Virtual Private Network*) ou Rede Privada Virtual é uma rede de comunicações privada que é construída em cima de uma que utiliza uma rede de comunicações pública (como por exemplo, a Internet). Para tornar a comunicação segura as VPNs usam protocolos de criptografia que formam um "túnel" entre a origem e o destino, impedindo que usuários não autorizados tenham acesso a essas informações.

Uma das maiores vantagens de utilizar o roteador é permitir que dois grupos de segmentos de redes se comuniquem entre si ao mesmo tempo em que elas continuam isoladas fisicamente. Nessa configuração, o roteador pode, também, funcionar como filtro de pacotes, possibilitando o controle de acesso à rede à qual ele está inserido. Funcionando como filtros, os roteadores podem ser inseridos como parte importante na política de segurança da rede, exercendo a função de *firewall*, filtrando pacotes e/ou protocolos específicos.

Por exemplo, um certo provedor internet disponibiliza alguns serviços a seus usuários tais como: HTTP (*web*), FTP (envio/recebimento de arquivos), SMTP (envio de mensagens), POP3 (recebimento de mensagem). Como parte de um nova política de segurança implantada, faz-se necessário bloquear o serviço de FTP, evitando um possível ataque externo mediante esse tipo de protocolo. Para fazer isso, basta configurar a roteador/*firewall* com uma regra impedindo a entrada e saída de requisições FTP. Depois da regra escrita e ativa, estará fechado o acesso aos usuários para envio e recebimento de arquivos via FTP.

Fixe os conhecimentos sobre roteador assistindo ao vídeo "Roteadores" disponível em http://www.youtube.com/watch?v=J3lWLlJs69M

4.8.4 Requisitos de um roteador

De acordo com Lages (2012), para um roteador funcionar de forma adequada é necessário que ele faça algumas tarefas:

- O roteador deve conhecer a topologia de suas porções de rede e escolher os caminhos adequados dentro delas.

- O roteador deve cuidar para que algumas rotas não sejam sobrecarregadas, enquanto outras fiquem sem uso.

- O roteador deve resolver os problemas que ocorrem quando a origem e o destino estão em redes diferentes.

4.9 Roteamento

De acordo com Comer (2007), roteamento se refere ao processo de selecionar um caminho pelo qual são enviados os pacotes, do *host* de origem até o *host* de destino remoto. Para que a comunicação entre esses dois *hosts* se efetive, os pacotes enviados devem passar por várias redes e, consequentemente, por vários roteadores. O caminho que o pacote vai passar é definido por cada roteador, também chamado de nó, que ele encontra pela frente. Esse caminho é definido por cada roteador baseado na sua tabela de roteamento. A tabela de roteamento é construída baseada em vários parâmetros, como rota menos congestionada, rota mais curta, rota com menos perda de pacote, etc. A união desses parâmetros é chamada de métrica. É a métrica menor que determina para qual dos enlaces físicos entre os roteadores o pacote será encaminhado.

4.9.1 Roteamento direto e indireto

Como já referido anteriormente, segundo Comer (2007), o roteamento pode ser divido em duas categorias: roteamento direto e indireto.

O roteamento direto acontece quando o *host* de destino está dentro da mesma rede do remetente. Nesse caso, o pacote recebe o endereço MAC da estação de destino.

O roteamento indireto acontece quando o *host* de destino está em uma rede diferente do remetente. Nesse caso, o pacote recebe o endereço MAC do *gateway* padrão e o endereço IP do destinatário. Esse *gateway* padrão verifica se o endereço do destinatário final pertence a uma das redes a ele conectadas. Em caso positivo, envia o pacote direto para a estação destinatária final, através de seu endereço MAC. Caso contrário, o pacote recebe o endereço MAC de outro *gateway* (de acordo com a tabela de roteamento do primeiro *gateway*) mais o endereço IP do destinatário. Esse segundo *gateway* fará a mesma verificação. O processo se repetirá até que o destinatário seja encontrado ou termine o tempo de vida do pacote.

4.9.2 Rotas estáticas e dinâmicas

Segundo Lages (2012), as tabelas de rotas podem conter rotas estáticas e/ou dinâmicas.

As rotas estáticas são criadas manualmente na tabela de roteamento, e uma vez criadas, não são mais alteradas. Esse método tem a vantagem de ser bastante simples, mas tem a desvantagem de, geralmente, levar à má utilização dos meios de comunicação, a não ser que o tráfego da rede seja bem regular

e bastante conhecido. No caso de ocorrência de falhas ou na ampliação da rede, é necessário reconfigurar manualmente as tabelas de roteamento.

Já as rotas dinâmicas são criadas dinamicamente na tabela de roteamento, através de comunicação entre os roteadores e de acordo com a carga na rede. As tabelas contêm as métricas de cada rota ou enlace e são consultadas pelo o roteador ou nó, para a escolha da melhor rota.

O inconveniente das rotas dinâmicas é que as tabelas devem ser periodicamente atualizadas (por protocolos de roteamento). De acordo com Soares (1995), essas atualizações podem ocorrer dos seguintes modos:

- No modo isolado, a atualização é realizada com base nas filas de mensagens para os diversos caminhos e outras informações locais.

- No modo distribuído, cada nó envia periodicamente aos outros nós, incluindo os *gateways*, as informações locais sobre a carga na rede. Essas informações são utilizadas para o cálculo da nova tabela.

- No modo centralizado, cada nó envia a um ponto central da rede as informações locais sobre a carga. Essas informações são utilizadas pelo ponto central para o cálculo das novas tabelas, que são então enviadas aos *gateways* e demais nós.

4.9.3 Roteamento centralizado

De acordo com Soares (1995), no roteamento centralizado existe, em algum lugar da rede, um Centro de Controle de Roteamento (CCR) responsável pelo cálculo das tabelas de rotas.

A utilização desse modo tem como vantagem o fato de o CCR sempre poder tomar decisões precisas sobre o caminho ótimo, uma vez que possui todas as informações da rede. No entanto, esse tipo de roteamento apresenta alguns problemas.

Um primeiro problema é o alto tráfego gerado na rede pela atualização das tabelas de roteamento, principalmente para uma rede com um grande número de nós. Se as mudanças forem frequentes, isso se torna um sério problema. No entanto, se ele não for tão frequente, o método torna-se bastante razoável. Um exemplo seria se as mudanças se restringissem à topologia da rede, que são em geral pouco frequentes; esse método seria útil.

Um segundo problema é a confiabilidade. Uma falha no CCR é crítica, pois, para toda a rede. Um aumento da confiabilidade, colocando outro CCR redundante, pode ser inviável pelo custo. Além disso, seria necessário algum protocolo para determinação de qual CCR deveria atuar em um dado instante.

Um quarto problema vem do fato de que os nós recebem suas tabelas em tempos diferentes, devido a retardos sofridos pelas mensagens que transportaram. Esse fato pode gerar inconsistências de roteamento, causando maiores retardos, inclusive nas mensagens que transportam as tabelas.

4.9.4 Roteamento isolado

De acordo com Soares (1995), no roteamento isolado a atualização é realizada com base nas filas de mensagens para os diversos caminhos e outras informações locais.

Um algoritmo de roteamento isolado simples é o apresentado por Baran (1989). Nesse algoritmo um nó ao receber um pacote tenta se livrar dele imediatamente pelo enlace que possui a fila mais curta no momento.

Para melhorar o algoritmo pode ser combinado com o encaminhamento por rota fixa. Usando o algoritmo, ao encaminhar um pacote, deve-se levar em conta não só a rota estática especificada, mas também o tamanho das filas. Por exemplo, até certo limiar do tamanho das filas, o pacote deve ser encaminhado para a menor fila; depois desse limiar, deve ser utilizada a rota estática.

Outro algoritmo de roteamento isolado também apresentado por Baran e Wu (1989), leva em conta que cada pacote deve incluir o nó de origem e um contador de saltos para contar os intermediários por onde trafegou. Ao receber um pacote, o nó sabe a que distância o nó de origem está, a partir do enlace de chegada. Se na sua tabela o nó de origem está a uma distância maior, ele deve atualizar a tabela. Passado um determinado tempo, cada nó possuirá o caminho mais curto para qualquer outro nó. Como cada nó só registra a troca para melhor, de tempos em tempos ele deve reiniciar o processo, de forma que enlaces que deixaram de ser bons, como por exemplo aqueles que ficaram sobrecarregados, não afetem a confiabilidade da tabela. Se as tabelas forem reiniciadas em um período pequeno, pacotes podem ter de ser transferidos por rotas desconhecidas. Se forem reiniciadas em um período longo, os pacotes podem ser transmitidos em rotas congestionadas ou com enlaces em falha.

Outro algoritmo de roteamento isolado é o que tem como base a exigência de que um pacote, ao chegar a um determinado nó da rede, seja enviado por todos os enlaces de saída desse nó, exceto por aquele por onde chegou. Esse algoritmo é simples, porém de grande custo devido à quantidade de pacotes gerados.

4.9.5 Roteamento distribuído

De acordo com Soares (1995), no modo distribuído, cada roteador envia, periodicamente, informações locais sobre a carga na rede aos outros roteadores a ele conectados. Essas informações são utilizadas para o cálculo da nova tabela.

Existem vários algoritmos para o roteamento distribuído. Será descrito apenas um deles, que é muito encontrado em inter-redes formadas por redes de difusão.

Nesse algoritmo, quando um roteador quer descobrir o melhor caminho, o de menor métrica, para enviar um pacote até um *host* de destino, ele envia um pacote de requisição por difusão (em todos os seus enlaces) na rede, com um campo com o endereço do nó de destino e um campo com um caminho, inicialmente com seu endereço. Cada nó intermediário, ao receber o pacote de difusão, verifica se o campo caminho contém seu endereço. Se não contiver, o nó acrescenta ao caminho seu endereço e o difunde por todos os seus enlaces. Em caso contrário, simplesmente descarta o pacote. O nó de destino vai receber vários pacotes, pelos diversos caminhos que possui, desde o nó de origem. O primeiro deles contém no campo caminho a rota de menor retardo. O nó de destino envia então, como resposta ao pacote recebido, outro pacote com essa rota (por exemplo, através da rota reversa ou uma rota que no momento lhe pareça a de menor retardo entre o destino/nova origem e a origem/novo destino).

De forma a manter sua tabela de rotas, um nó deve periodicamente enviar pacotes de difusão para descobrimento destas. Quanto menor o período, melhor será a adaptação do nó às flutuações de tráfego e o algoritmo. Quanto maior o período, menor é o tráfego que o algoritmo gera na rede. Uma solução de compromisso deve ser encontrada.

4.9.6 Roteamento hierárquico

De acordo com Soares (1995), quando as redes tornam-se muito grandes, o número de entradas na tabela de rotas, para qualquer um dos roteamentos descritos anteriormente, pode ser tão elevado que as tornam impossíveis

de serem armazenadas ou percorridas. A solução nesses casos é realizar o roteamento hierarquicamente.

No roteamento hierárquico os nós são divididos em regiões, com cada nó sendo capaz de manter as informações de rotas das regiões a que pertence. A subdivisão da camada de rede no modelo RM-OSI é denominada região. Essa hierarquia em dois níveis (rede e região) pode ser insuficiente para redes muito grandes, podendo ser necessário agrupar as regiões em super-regiões, e assim sucessivamente.

4.9.7 Tabela de roteamento

Segundo Comer (2007), o roteamento é feito com base numa tabela. Essa tabela de roteamento fica armazenada em um nó, ou roteadores. Ela contém informação sobre os possíveis destinos a partir desse nó. Ela possui a informação de qual é o endereço do *gateway* padrão para uma dada rede.

Uma tabela de roteamento é composta das seguintes informações:

- Todos os endereços de rede conhecidos.
- O endereço dos *gateways* para alcançar essas redes.
- Instruções para conexão as outras redes.
- Os caminhos possíveis entre os roteadores.
- O custo do envio dos dados sobre tais caminhos.

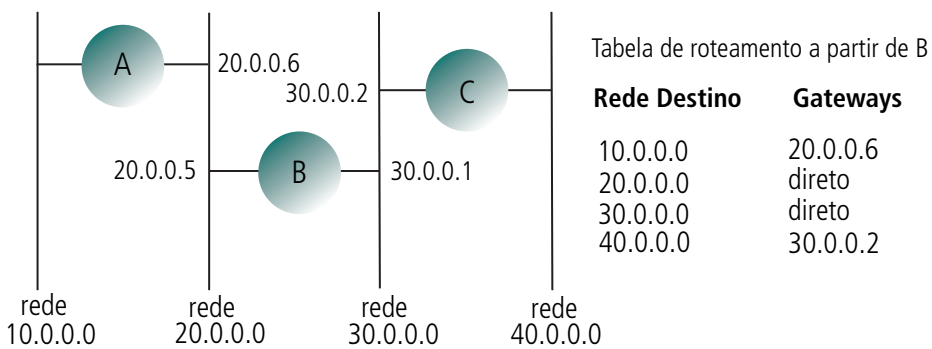


Figura 4.29: Exemplo de tabela de roteamento
Fonte: http://pt.scribd.com/doc/30413789/Apostila-Redes-Com-Put-Adores-Lages

Outro exemplo de construção da tabela de roteamento pode ser visto na Figura 4.30, na qual estão representados um sistema de sub-redes com seus roteadores e os respectivos enlaces entre esses roteadores, e na Tabela 4.4,

que mostra um comparativo de roteamento que o roteador R1 construiu pelo critério de menor métrica, que significa menor custo.

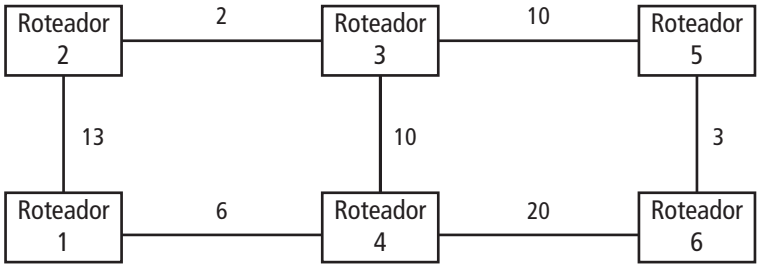


Figura 4.30: Enlaces entre roteadores com respectivos custos

Tabela 4.4: Roteamento do roteador R1

DESTINATÁRIO	PRÓXIMO HOP	CUSTO
RI	RI	Ligação Local
RI	R2	13
RI	R3	15
RI	R4	6
RI	R5	25
RI	R6	26

Fonte: Elaborada pelo autor

4.10 Métrica

Segundo Castro et al. (2002), métrica é o padrão de medida que é usado pelos algoritmos de roteamento para determinar o melhor caminho para um destino. Pode-se utilizar apenas um parâmetro ou vários parâmetros. A utilização de vários parâmetros permite uma melhor modelagem da métrica e uma decisão mais eficiente de qual é o melhor caminho.

Alguns parâmetros utilizados para construir a métrica são:

- tamanho do caminho;
- confiabilidade;
- atraso;
- largura de banda;
- carga;
- custo da comunicação.

4.11 Algoritmos de roteamento

De acordo com Castro *et al*. (2002), o algoritmo de roteamento define como a rede deverá ser mapeada para a construção da tabela de roteamento dos nodos, ou roteadores, que fazem parte dessa rede. Por isso, um algoritmo de roteamento deve proporcionar:

- correção;
- simplicidade;
- robustez;
- estabilidade;
- consideração com o usuário;
- eficiência global.

Baseado nessas premissas, pode-se classificar o algoritmo em duas classes: o *distance vector* e o *link-state*.

4.12 Algoritmos – *Distance Vector*

De acordo com Castro et al. (2002), esse algoritmo é bastante simples e baseia-se na distância entre dois pontos. Essa distância refere-se ao número de *gateways* (ou número de roteadores) existentes na rota utilizada, sendo medida em *hops*, que são a passagem de um datagrama por cada *gateway* (ou roteador).

Os roteadores que usam algoritmos do tipo *distance vector* mantêm apenas uma tabela de roteamento com as rotas para os roteadores vizinhos que são conhecidas pelo roteador. Esses roteadores trocam periodicamente informações a respeito das suas tabelas de roteamento, mesmo que elas não tenham sido alteradas desde a última troca de informações.

> Ele leva em conta o número de saltos da rota e a distância administrativa para encaminhar um pacote. O número máximo de saltos é 15. Assim, quando há uma rota com métrica 16, isso significa que aquela rota está inutilizável. Quando ocorre uma atualização na tabela, toda a tabela é divulgada aos demais roteadores. Essas atualizações se dão em *broadcast*. Quando uma rota é aprendida através de um roteador vizinho, assume-se que a rota é através daquele roteador. Nesse esquema, o roteador não conhece a topologia. Somente as sub-redes diretamente conectadas são conhecidas pelo roteador (LAGES, 2012).

4.13 Algoritmos – *Link-State*

Segundo Castro et al. (2002), os algoritmos do tipo *link-state* geram e mantêm um mapa lógico de toda a rede. Essa manutenção do mapa lógico é realizada pelo envio, por um roteador *links-tate*, de um pacote com informações sobre todos os seus enlaces (conexões para redes e conexões para outros roteadores) para todos os outros roteadores *link-state* existentes na rede. Esse procedimento é chamado de *flooding*. Cada roteador usa essas informações para construir um mapa da rede, que é concluído quando todos os roteadores atingirem o mesmo mapa da rede. Os roteadores *link-state* somente retransmitem informações entre si quando ocorrer uma mudança na rota ou serviço. Veja na figura 4.31, e na tabela 4.5, exemplos de roteamento de algoritmos *link-state* e *distance vector.*

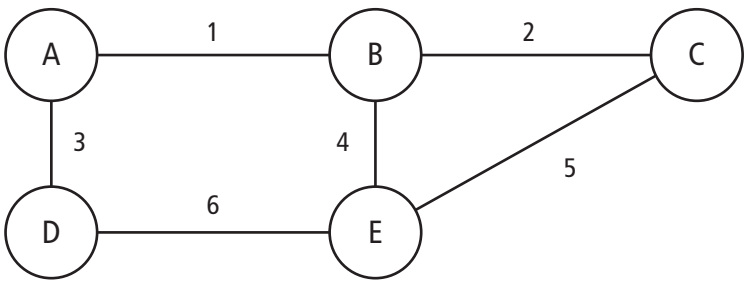


Figura 4.31: Rede com roteadores
Fonte: Elaborada pelo autor

Tabela 4.5: Roteamento

Destino	Ligação	Distância	Fonte	Ligação	Distância
A	Local	0	A	B	1
B	1	1	B	C	2
C	1	2	A	D	3
D	3	1	B	E	4
E	1	2	C	E	5
			E	D	6
Roteamento *distance- vector*			Roteamento *link-state*		

Fonte: Elaborada pelo autor

4.14 *Exterior Gateway Protocol* (EGP)

Um sistema autônomo é composto por uma ou mais redes e é administrado por uma única entidade. O sistema autônomo tem livre escolha do protocolo a ser utilizado para descobrir, manter, divulgar e atualizar rotas dentro do seu universo.

De acordo com Comer (2007), dois roteadores que trocam informações sobre roteamento são considerados vizinhos externo se pertencem a dois sistemas autônomos diferentes. Para se comunicarem, eles utilizam o protocolo EGP (*Exterior Gateway Protocol*). Cada sistema autônomo escolhe alguns roteadores para intermediar a comunicação com o mundo externo. Esses roteadores são denominados "roteadores externos".

Os roteadores externos tornam-se vizinhos EGP. Os vizinhos EGP traçam informações sobre as redes que podem ser alcançadas no interior dos seus respectivos sistemas autônomos.

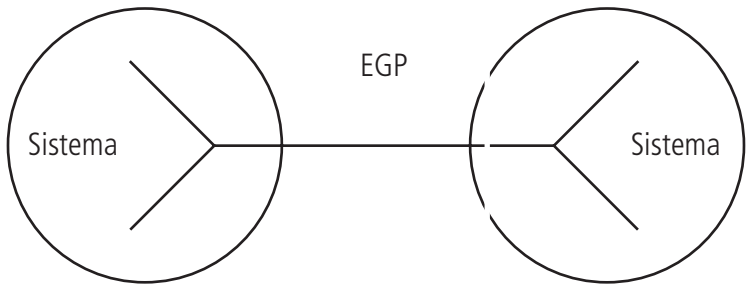


Figura 4.32: Conceito de vizinhança
Fonte: Elaborada pelo autor

O protocolo EGP possui três características principais:

- suporta mecanismo de aquisição de vizinho;
- faz testes contínuos para ver se os vizinhos estão respondendo;
- Divulga informação entre vizinhos utilizando mensagens de atualização de rotas.

4.15 *Interior Gateway Protocol* (IGP)

De acordo com Comer (2007), roteadores que trocam informações de roteamento somente com roteadores do mesmo sistema autônomo são considerados "vizinhos interiores" e utilizam diversos protocolos denominados genericamente IGP (*Interior Gateway Protocols*). Entre eles encontra-se o RIP (*Routing Information Protocol*), o HELLO, o OSPF (*Open Shortest Path First*), o IGRP (*Internal Gateway Routing Protocol*).

4.15.1 RIP (TCP/IP)

Segundo Torres (2007), o *Routing Information Protocol* (RIP) é um protocolo IGP que usa o algoritmo *distance vector*, por isso ele transmite periodicamente uma mensagem de atualização de roteamento (de 30 em 30 segundos) para cada rede que ele pode alcançar, repassando o custo de acesso a elas. Essas mensagens são enviadas via *broadcast* para os roteadores RIP.

Cada mensagem de atualização de roteamento recebida é incorporada à tabela de roteamento dos roteadores alcançados. O roteador que enviou a mensagem é identificado como o próximo roteador (*next hop router*) na rota do roteador que recebeu a mensagem.

Se um roteador for informado da existência de duas rotas de acesso a uma mesma rede, ele guarda na tabela de roteamento apenas o próximo *hop* e a distância da rota de caminho mais curto. O RIP usa como distância a contagem dos *hops*, ou seja, o número de roteadores existentes no caminho até o destino. O RIP do TCP/IP limita a distância em 15 *hops* ou saltos. Se uma rota tiver 16 ou mais saltos ela é automaticamente descartada.

A partir do momento em que uma rota é memorizada na tabela de roteamento, sua existência precisa ser verificada a intervalos regulares. Os roteadores RIP normalmente transmitem uma mensagem de atualização de roteamento com todas as rotas a cada 30 segundos.

Sempre que uma rota é atualizada, em consequência de uma mensagem de atualização de roteamento, dispara-se um temporizador. Se não for recebida nenhuma outra mensagem de atualização dessa rota, dentro de 180 segundos, ela é considerada inativa em decorrência de uma falha de rede ou de nó (métrica 16 usada para esse fim), retirada da tabela de roteamento e divulgada para os vizinhos por 120 segundos.

Informações guardadas na tabela de roteamento do RIP (CASTRO *et al.*, 2002):

- Endereço de destino;
- Endereço do próximo roteador;
- Interface do *host* a ser utilizada;
- Métrica da rota;
- *Flags* e *timers* que controlam tempos de atualização.

O protocolo RIP tem a desvantagem de não trabalhar com máscara de sub-redes; dessa maneira, ele só pode interligar redes que trabalhem com endereços IP classes *full* e não conseguem rotear endereços CIDR. Quando o RIP é usado em uma rede com sub-redes, todas as sub-redes são forçadas a usar a mesma máscara.

4.15.2 RIP II (TCP/IP)

De acordo com Torres (2009), o RIP II é um aprimoramento do RIP, incluindo a máscara de sub-rede nas suas rotas. Assim o RIP II pode ser usado em topologia de rede que exijam o uso de máscaras de sub-rede com comprimentos variáveis, podendo suportar a sub-rede zero. Outra vantagem do RIP II é o suporte a autenticação de pacotes por *password*, evitando a adulteração destes no trajeto pela rede.

4.15.3 *Open Shortest Path First* (OSPF)

OSPF é um protocolo de roteamento IGP do tipo *link-state* que, segundo Torres (2009), faz parte do conjunto de protocolos TCP/IP. Os roteadores do tipo *link-state* trocam informações sobre a topologia da rede, incluindo o estado de funcionamento de cada enlace e a distância associada entre seus roteadores. A partir desses dados trocados, cada roteador constrói o seu mapa da rede, que o utiliza para a extração dos dados necessários para o roteamento.

Para cada destinatário, os roteadores OSPF consultam a sua base de dados de *link-state* e selecionam a rota que proporcione o caminho mais curto. Na sequência, as informações de *link-state* são compartilhadas com outros roteadores em diferentes áreas de acordo com a relação que guardam entre si.

Os roteadores que utilizam o OSPF conseguem tomar decisões de roteamento com base nos seguintes parâmetros (CASTRO et al., 2002):

- carga de tráfego;
- ***throughput***;
- custo do circuito;
- prioridade de serviço atribuída aos pacotes que se originam ou se destinam a um ponto específico.

A-Z

Throughput
(ou taxa de transferência)
É a quantidade de dados transferidos de um lugar a outro.

Para fins administrativos, a rede OSPF pode ser subdividida em varias regiões ou áreas. Todos os roteadores da mesma área trocam entre si todas as informações sobre o estado completo dos enlaces. As informações trocadas entre os roteadores de áreas diferentes consistem apenas em um resumo da topologia.

4.15.4 RIP, RIP II e OSPF

Conforme vimos nas descrições anteriores dos protocolos RIP, RIP II e OSPF, podemos constatar que a grande vantagem dos protocolos RIP e RIP II é que eles são simples de ser configurados. Por isso eles são implementados pela grande maioria de roteadores. A diferença entre o RIP e RIP II é que o RIP, por não trabalhar com máscara de sub-rede, é recomendado apenas para pequenas redes (algo em torno de 40 ou 50 nós no máximo).

O protocolo OSPF, apesar de ter configuração mais complicada e apresente maior dificuldade inicial de aprendizado, possui as vantagens de possibilitar organização hierárquica, redução de *overhead*, convergência das rotas mais rápida e maior tolerância à falhas. Ele é também um protocolo seguro quanto a ataques de informações de roteamento. Devido a essas características, o protocolo OSPF é ideal para ambientes mais complexos ou com previsão de crescimento.

Tabela 4.6: Comparação entre protocolos

Característica	Protocolo		
	RIP I	**RIP II**	**OSPF**
Autenticação dos pacotes	Não	SIM (opcional com *password*)	SIM (opcional com *password* ou MD5)
Inclui máscara de sub-rede	Não	Sim	Sim
Frequência de mensagens de atualização de rotas enviadas	A cada 30 segundos (independentemente de mudanças)	Mesmo que RIP	Só em casos de mudanças
Propagação de rotas na rede (tempo de convergência)	Lenta	Lenta	Rápida
Consumo de banda na propagação de rotas	Alto	Alto	Baixo
Organização da rede	*Flat*	*Flat*	Conceito de Áreas e Fronteiras
Complexidade na configuração e administração da rede	Pequena	Pequena	Grande
Uso de IP *multicast* no envio de atualização de rotas	Não	Sim	Sim

Fonte: Elaborada pelo autor

4.15.5 Hello

O algoritmo *vector distance*, segundo Comer (2007), tem um emprego diferente neste protocolo, que é medir as distâncias entre redes por tempo e não por saltos. Possui duas funções básicas: manter o sincronismo dos relógios de todos os roteadores envolvidos e divulgar alcance por tempo referente a cada rede.

4.16 *Gateway* X *Switching* X *Routing* X *Bridging*

Como foi visto, cada estratégia de segmentação de redes possui suas características próprias, que determinam sua maior ou menor adequação para determinada rede local e para os problemas que essa rede apresenta. Muitas vezes, várias técnicas são adequadas e implementadas. Em outras, devido a limitações impostas pelas características da rede, apenas uma é indicada.

Normalmente, *gateway* conversor de protocolo deve ser usado apenas quando é necessária a conversão de protocolos, pois sua configuração é difícil, e o custo elevado. Eles trabalham na camada 7 do modelo RM-OSI.

Por sua vez, os roteadores são indicados para a conversão de meios e trabalham na camada 3 do modelo RM-OSI. O roteador tem a finalidade de encaminhar, inteligentemente, pacotes entre os meios que ele interliga, descobrindo quando deve fazer o encaminhamento e qual o melhor caminho que deve ser utilizado, otimizando o tráfego entre redes, além de prover mecanismos para controle de fluxo. O principal uso dos roteadores está na interligação de redes LAN com interligação para redes WAN, pois eles filtram os pacotes de *broadcasts*. Um grande problema dos roteadores é que eles exigem muitas configurações, pois se faz necessário selecionar um protocolo de roteamento e nomear ou numerar as redes com identificadores únicos.

Os *switches* camada 2 do modelo OSI só entendem endereços MAC. Dessa maneira, eles sabem apenas levar o pacote de uma porta a outra, sem tratamento de rota, já que reconhecem somente o caminho entre suas portas, não podendo interligar meios diferentes. A instalação dos *switches* é simples, especialmente considerando que eles descobrem novos *hosts* interligados em suas portas, através de seus endereços MAC, sem intervenção manual.

Porém, a análise da melhor aplicação para cada estratégia é bastante complexa, levando em consideração fatores gerais da rede e exigindo bom conhecimento dela. Como foi dito anteriormente, em algumas redes todas as estratégias são adequadas; em outras, por sua vez, apenas uma estratégia serve como solução.

Camada	Equipamento	Características
Aplicação	Gateway	Traduz um protocolo em outro (SNA, IPX, TCP/IP, etc)
Transporte	----------	
Inter-Rede	Roteador	Examina e despacha pacotes de acordo com o endereço de destino final
Interface de Rede	Ponte	Podem traduzir protocolos de acesso (Ethernet, Token-Ring, etc)
Intra-Rede	Repetidor	Permite mudar o meio físico

Figura 4.33: Camada de atuação dos elementos ativos de rede
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-redes-ip/RedesIP-Cap-01.pdf

4.17 Domínios de colisão X domínios de *Broadcast*

Segundo Gonçalves (2012), a transmissão de pacotes em uma rede Ethernet sempre ocorre em *broadcast*. Nesse modo de transmissão, o pacote é difundido por toda a rede para todos os *hosts* ao mesmo tempo. Essa difusão pode ocasionar colisões, pois dois ou mais *hosts* podem tentar transmitir ao mesmo tempo e somente um poderá fazê-lo por vez, caracterizando uma competição entre os *hosts*, pelo meio físico de transmissão.

Ainda de acordo com Gonçalves (2012), nesse panorama, o domínio de colisão é definido como uma mesma rede LAN interligando vários *hosts* diretamente, por meio físico compartilhado ou através de concentradores que atuam na camada 1 do modelo OSI, como os *hubs* por exemplo. Isso ocorre porque nessa situação não há segmentação da rede, havendo competição pelo mesmo meio físico. As ocorrências de colisões aumentam à medida que a rede cresce. E todas as vezes que há uma colisão a rede interrompe seu tráfego, para tratá-la. Essas paradas trazem perda de *performance* da rede. Esse problema é facilmente eliminado por concentradores que trabalham na camada 2 do modelo RM-OSI como os *switches*.

Já o domínio de *broadcast* é definido nas fronteiras de alcance de um pacote de *broadcast*. Pacotes de *broadcast* também diminuem a *performance* das rede que ele alcança, pois todos os *hosts* desse domínio devem processá-lo. Assim, toda vez que um pacote de *broadcast* é enviado, os *hosts* devem parar de transmitir para tratá-lo. Esse problema é facilmente eliminado por concentradores que trabalham na camada 3 do modelo RM-OSI como os roteadores.

Resumindo:

- Repetidor/*hub*: os domínios de colisão e *broadcast* de todos os segmentos de rede interconectados por eles são os mesmos.

- Ponte/*switch*: os segmentos de rede, ou *hosts* interconectados por eles, possuem o mesmo domínio de *broadcast* (mesma rede); porém, os domínios de colisão são separados, pois há uma segmentação da rede (não propaga colisão entre os segmentos).

- Roteador: este equipamento de interconexão de redes isola totalmente as redes por ele conectadas, ou seja, tanto o domínio de colisão quanto o de *broadcast* são diferentes.

Para saber mais sobre como montar uma rede sem fio, acesse http://www.clubedohardware.com.br/artigos/Como-Montar-uma-Rede-Sem-Fio-Usando-um-Roteador-de-Banda-Larga/1331

Acesse o Tutorial Rede Wireless em http://www.babooforum.com.br/forum/index.php?/topic/269602-tutorial-redes-wireless-%26gt%3B%26gt%3Batualizado%26lt%3B%26lt%3B/

Veja "Conexões Wireless" em http://olhardigital.uol.com.br/produtos/central_de_videos/veja-dicas-para-melhorar-sua-conexao-wireless

Veja como montar uma rede em cinco minutos em http://www.youtube.com/watch?v=n03AGNOjd1o

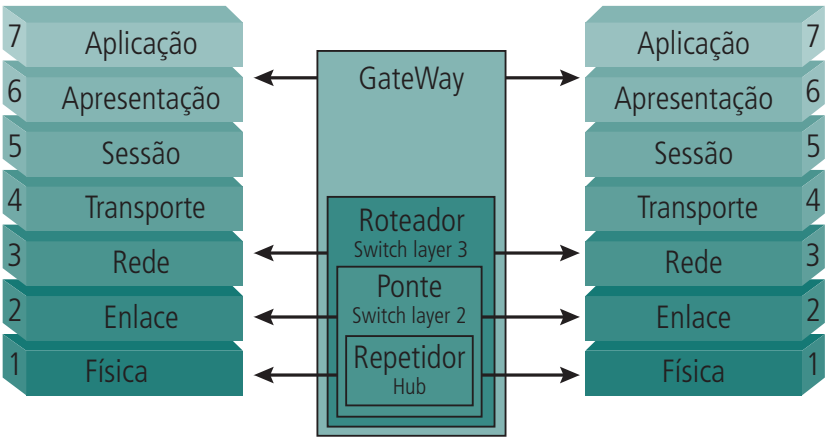


Figura 4.34: Camada de atuação dos elementos ativos de rede no modelo OSI
Fonte: http://www.cesarkallas.net/arquivos/faculdade-pos/TP311-Redes-IP/RedesIP-Cap-01.pdf

Resumo

Veja o vídeo sobre elementos ativos de rede, disponível em http://www.youtube.com/watch?v=psbSuIBJ-1w. Faça um estudo comparativo e poste no AVEA.

Nesta aula você conheceu os principais elementos de conectividade para redes de computadores, suas características e funcionamentos e fez um estudo comparativo entre esses equipamentos. Viu que o roteador é um equipamento que permite interligar redes fisicamente separadas e que foram eles que fizeram a internet chegar ao que é hoje. Assim, estudou como se dá o roteamento, que são feitos e gerenciados por algoritmos e protocolos. Aprendeu que além de existirem as redes fisicamente separadas, pode-se isolá-las virtualmente através de VLANs e, por último, aprendeu a diferença entre domínio de colisão e domínio de *broadcast*.

Atividades de aprendizagem

1. Responda as questões a seguir e poste no AVEA.

a) Cite os principais protocolos de roteamento utilizados na arquitetura TCP/IP.

b) Descreva as formas de roteamento existentes e o seu funcionamento.

c) Como funciona o roteamento *distance vector*?

d) Como funciona o roteamento *link-state*?

e) Suponha que um datagrama passa por n roteadores em uma viagem através de uma inter-rede. Quantas vezes o datagrama é encapsulado?

f) Quais são as diferenças entre *hub* e roteador? E quais são as vantagens de um em comparação ao outro?

g) Como funciona um repetidor?

h) Podemos afirmar que o *hub* tem uma função lógica e outra física em redes de computadores?

i) Quais os tipos de *switch* que podemos encontrar?

j) Faça uma comparação entre *switch* e *hub*.

k) Descreva os protocolos de roteamento interno.

Aula 5 - Segurança da Informação

Objetivos

Identificar os processos e sistemas de segurança dentro do contexto de uma visão ampla de políticas de segurança.

Conhecer os risco e vulnerabilidades dos sistemas computacionais.

Conhecer as técnicas e mecanismos de defesa.

> "Assim como os pertences precisam de cadeados para que fiquem seguros, os computadores e redes de dados necessitam de dispositivos para garantira segurança da informação" (COMER, 2007).

Para Laureano (2005), a segurança da informação é a proteção dos sistemas de informação contra a negação de serviço a usuários autorizados, assim como contra a intrusão, e a modificação não autorizada de dados ou informações, armazenados, em processamento ou em trânsito, abrangendo a segurança dos recursos humanos, da documentação e do material, das áreas e instalações das comunicações e computacional, assim como as destinadas a prevenir, detectar, deter e documentar eventuais ameaças a seu desenvolvimento.

Com o advento da internet, a segurança de dados tornou-se um processo crítico, pois os padrões da internet são abertos e de domínio público; e não foram projetados tendo como primícias a segurança. Aliás, o objetivo era descentralizar a informação, tornando mais fácil o seu acesso. Mas pessoas e organizações mal-intencionadas podem interceptar e transmutar mensagens e dados.

A segurança de uma determinada informação pode ser afetada por fatores comportamentais e de uso, pelo ambiente ou infraestrutura que a cerca ou por pessoas mal-intencionadas que têm o objetivo de furtar, destruir ou modificar tal informação. Dessa maneira, para a preservação da integridade dos dados, uma série de medidas e políticas deve ser adotada a fim de preservá-los, quer seja uma proteção lógica ou física.

Uma política específica é que ira definir o nível de segurança. Mas é necessário que a política seja seguida pela organização ou pessoa, para garantir que uma vez estabelecidos os princípios, aquele nível desejado seja perseguido e mantido. Mas vale ressaltar que é necessário dosar o nível a ser adotado, pois quanto maior esse nível, , mais difícil fica o uso da informação. A ideia é definir, de modo equilibrado, que nível de segurança *versus* usabilidade é o ideal para determinada empresa ou pessoa.

Uma boa política de segurança deve estar baseada em atributos que orientem a sua análise, seu planejamento e a sua implementação. E um parâmetro muito usado para isso, segundo Lauereano (2005), é o trio definido como CIA (*Confidentiality Integrity and Availability* – Confidencialidade Integridade e Disponibilidade). A definição do trio CIA, segundo os padrões internacionais, é:

- **Confidencialidade**: propriedade que limita o acesso à informação tão somente às entidades legítimas, ou seja, àquelas autorizadas pelo proprietário da informação.

- **Integridade**: propriedade que garante que a informação manipulada mantenha todas as características originais estabelecidas pelo proprietário da informação, incluindo controle de mudanças e garantia do seu ciclo de vida (nascimento, manutenção e destruição).

- **Disponibilidade**: propriedade que garante que a informação esteja sempre disponível para o uso legítimo, ou seja, por aqueles usuários autorizados pelo proprietário da informação.

Outros atributos, igualmente, importantes são a irretratabilidade e a autenticidade. Com a evolução do comércio eletrônico e da sociedade da informação, o atributo privacidade passou, também, a ser uma grande preocupação.

Mas é importante notar que um sistema nunca está totalmente seguro, pois à medida que os sistemas de segurança se sofisticam, também o fazem as técnicas de invasão. O segredo é estar sempre um passo à frente. Podem ser estabelecidas métricas (com o uso ou não de ferramentas) para a definição do nível de segurança existente e, com isso, serem estabelecidas as bases para análise da melhoria ou piora da situação de segurança existente. Baseados nessas primícias é que vamos estudar políticas de segurança.

A política de segurança é um mecanismo preventivo de proteção dos dados e processos importantes de uma organização que define um padrão de segurança a ser seguido pelo corpo técnico e gerencial e pelos usuários, internos ou externos. Pode ser usada para definir as interfaces entre usuários, fornecedores e parceiros e para medir a qualidade e a segurança dos sistemas atuais (LAUREANO, 2005).

5.1 Definições dos atores locais

Segundo Nakamura (2007), os atores locais, recursos computacionais ou pessoas que compõem a rede local devem estar envolvidos na segurança.

5.1.1 *Site*

Define-se como *site* qualquer organização que possua computadores ou outro recurso de rede. Esses recursos podem ser servidores, roteadores, terminais, PCS ou outros dispositivos que tenham acesso à internet ou a uma intranet.

5.1.2 Administrador

O administrador pode ser uma pessoa ou um grupo de pessoas responsáveis pela operação cotidiana do sistema e dos recursos da rede. Existem dois elementos básicos nesse grupo, que são os administradores de segurança, que são os responsáveis pela segurança das informações ou da tecnologia de informática, e o gerente administrativo que é uma pessoa ou um grupo de pessoas em um *site* que definem ou aprovam uma política.

5.2 Invasores digitais

De acordo com Nakamura (2007), existem diversos tipos de invasores, desde aqueles que roubam a informação até os que destroem o sistema, passando pelos que só invadem por puro prazer. A seguir serão apresentados alguns desses invasores.

5.2.1 *Hacker*

Os *hackers* usam sua inteligência para o bem e não visam prejudicar ninguém, promovem invasões apenas pra provar sua capacidade, entrando e saindo sem causar nenhum dano, mas sempre deixando um alerta ao dono do *site* ou do serviço invadido.

5.2.2 *Crackers*

Os *crackers* invadem sistemas para deixar a sua marca, tirá-los do ar ou destruí-los por completo, ou usar as informações nele contidas. Geralmente, *crackers* são *hackers* que querem se vingar de alguém ou se aproveitar de algo; por isso ambos costumam sempre estar em conflito. Hoje, para se proteger, as grandes empresas, contratam *hackers* para ajudá-las na proteção de seus sistemas.

5.2.3 *Script kiddies* ou lammers

Veja a diferença entre *hackers* e *crackers* em http://olhardigital.uol.com.br/produtos/central_de_videos/hackers-e-crackers-%96-as-diferencas. Faça um estudo comparativo e poste no AVEA

Um *script kiddie* (garoto dos *scripts*, numa tradução literal) é um termo depreciativo atribuído aos grupos de *crackers* inexperientes (geralmente das camadas etárias mais novas) que usam o trabalho intelectual dos verdadeiros *crackers*. Não possuem conhecimento de programação, e não estão interessados em tecnologia, e sim em ganhar fama ou outros tipos de lucros pessoais.

5.2.4 *Newbie*

Newbie, que no inglês significa novato, indica uma pessoa aprendiz na área, ainda sem muita habilidade, porém com uma sede de conhecimento notável.

5.2.5 *Carders*

São aqueles *crackers* especialistas em roubos de número de cartões de crédito que, obviamente, utilizam esses números para fazer compras pela internet. Eles afetam tanto usuários comuns quanto empresas e são extremamente difíceis de localizar.

5.2.6 *Phreaker*

O termo *phreaker* vem do inglês *freak*, que significa "maluco". Ele é essencialmente um decifrador aplicado à área de telefonia (móvel ou fixa).

5.2.7 Funcionários

O fator humano é considerado por especialista, em segurança da informação, como uma das maiores causas de invasões e ataques aos sistemas de uma organização e um funcionário insatisfeito, desmotivado e desvalorizado é uma grande vulnerabilidade. O agravante desse panorama é que o ataque que vem de dentro da organização tornando difícil a sua previsão. E apesar de não serem especialistas em invasão, eles podem causar graves problemas utilizando do conhecimento adquirido dentro da própria empresa.

5.3 Técnicas de invasão

Normalmente, as várias técnicas básicas de invasão exploram problemas gerados pela má configuração de computadores, servidores em rede e equipamentos de interconexão de rede. Os diferentes ambientes de rede exigem diferentes abordagens na invasão. A abordagem usada na invasão de uma rede corporativa de maior porte será completamente diferente da abordagem usada em uma pequena rede que talvez nem esteja conectada diretamente à internet, como também será diferente da abordagem usada para invadir um usuário apenas.

Em termos de facilidade, uma rede pequena, que não tem contato com a internet, é a mais vulnerável, numa abordagem de dentro para fora, ou seja, a sua invasão é normalmente ocasionada por um funcionário. Contudo, tentar invadir uma rede dessas é muito difícil, pois não existem conexões permanentes com a internet. Nesses casos, ou o invasor está dentro da rede ou tentará comprometer qualquer computador que esteja na rede local, mas que possua algum tipo de acesso à internet.

Nesses casos, técnicas de engenharia social são muito usadas, pois a falta de conexão permanente limita muito a gama de ferramentas que podem ser usadas para extrair informações.

Redes que acessam a internet através de canais permanentes e que possuem servidores também conectados nessa estrutura, com endereços reais, disponibilizando serviços, são as mais vulneráveis, necessitando, segundo CERT.br (2012), de um nível de segurança bem elevado.

A seguir serão mostrados os tipos de ataques mais comuns que os sistemas computacionais podem sofrer.

5.3.1 *Probing*

Nesse ataque os invasores tentarão investigar a rede para determinar: que serviços rodam em que servidores; quais são as versões desses serviços; quais são os servidores e onde estão localizados na rede. Para tanto eles necessitam traçar um esboço ou um mapa da rede; conhecer as relações de confiança entre os servidores; os sistemas operacionais utilizados; as possíveis estações de gerência na rede; como é feita a filtragem de pacotes (se existir); os sistemas de detecção à intrusão – IDS (se existirem); os *honeypots* (se existirem); os *portscanning* (passivo e com *spoofing* se possível).

Dependendo da capacidade do invasor, a fase de *probing* será realizada por meio de algum método que impossibilite sua identificação, como provedores gratuitos ou acessos físicos de conexões roubadas.

5.3.2 Engenharia social

Essa técnica é utilizada para descobrir informações pessoais sobre os usuários da rede e da organização através de estudos simples sobre a vida deles.

Essas informações podem ser sobre fornecedores de suprimentos e manutenção; sobre as pessoas que acessam a rede e qual seu privilégio de acesso, e qual o seu grau de conhecimento sobre a rede, pois quanto menor, melhor, principalmente se possuir acesso privilegiado; sobre números de telefones importantes, tais como do administrador da rede, das pessoas envolvidas com a administração da infraestrutura de rede, telefones de departamentos como comercial, entre outros; consiste também em tentar obter uma lista de endereços de correio eletrônico importantes ou informações do suporte telefônico da empresa, caso possua.

Uma fonte importante de informações para os invasores que utilizam essas técnicas é o acesso ao lixo da vítima, pois muitas delas escrevem informações preciosas em papéis e, em vez de destruí-los depois, simplesmente os jogam no lixo.

Outra fonte importante está no alto escalão das organizações, que não possui muito conhecimento técnico e normalmente tem acessos privilegiados. É um setor que facilmente torna-se vítima de "cavalos de Troia" (ver seção 5.3.3) que vêm embutidos em *e-mails* falsos de cunho social ou como se fossem de conhecidos.

Uma vez instalados no computador invadido o Cavalo de Tróia passa a enviar informações ao invasor, e baseado nessas informações, ele irá pesquisar na internet e na sua comunidade sobre vulnerabilidades existentes nas versões dos programas, serviços e sistemas operacionais usados pela organização em foco. Além disso, caso a relação da rede interna com a rede de gerência seja direta, ele terá acesso a praticamente toda a rede dessa organização.

O invasor poderá usar o mesmo sistema, indo diretamente ao usuário da rede interna, como por exemplo, os funcionários de setores estratégicos como os departamentos administrativos, comerciais, e financeiros. Como

no caso anterior, a maioria dos funcionários desses departamentos é leiga e com frequência, também, abrem anexos de seu correio eletrônico que pode conter um Cavalo de Tróia.

É bem provável que, com alguns dias de investigação do tráfego da rede interna, o invasor consiga alguma senha com direitos de administração. Como administradores de rede tem o hábito de usar a mesma senha para diversas ferramentas, se na primeira fase alguma ferramenta de gerência remota foi achada, então, é mais do que provável que as senhas serão idênticas.

5.3.3 *Trojans*

O nome *trojan* é uma alusão à história do antigo Cavalo de Troia, em que o governante da cidade de Troia, na antiga Grécia, foi presenteado com um cavalo de madeira no qual havia soldados inimigos escondidos. Por analogia, os <*malwares* (programas maliciosos) conseguem ficar escondidos em arquivos de inicialização do sistema operacional e agem toda vez que a máquina é ligada.

Normalmente os *trojans* são programas que demonstram um determinado tipo de comportamento ou se propõem a uma determinada tarefa, Eles geralmente a realizam, porém, sem que o usuário saiba. Esta segunda função na maioria das vezes abre o computador para invasões ou acesso remotos.

Hoje em dia, existem inúmeros programas do tipo *trojan horse*, ou cavalo de Troia, mas o conceito aplicado à informática existe há décadas. O primeiro programa usado como *trojan horse* que ganhou a comunidade foi o NetBus. Após o NetBus, que é tido como um *software* de gerência remota, e não como um *trojan horse*, surgiram diversos outros, sendo o mais famoso deles, o Back Orifice.

5.3.4 *Backdoors*

Os *backdoors* podem ter mais ou menos a mesma funcionalidade de um *trojan*, mas com intenções. Quando um invasor consegue acesso a um sistema, uma de suas primeiras atitudes será instalar *backdoors* naquele local. Essas *backdoors* lhe permitirão voltar a ter acesso a este sistema se por acaso o dono / usuário ou administrador descobrir que sua segurança foi violada. Um *backdoor* pode estar na forma de um programa, assim como os *trojans*, como um *script* (principalmente em ambiente UNIX), ou até como uma série de procedimentos (criar uma conta com direitos de administração, com um nome comum). Essa é a principal diferença em relação a um *trojan*, que geralmente é um arquivo executável.

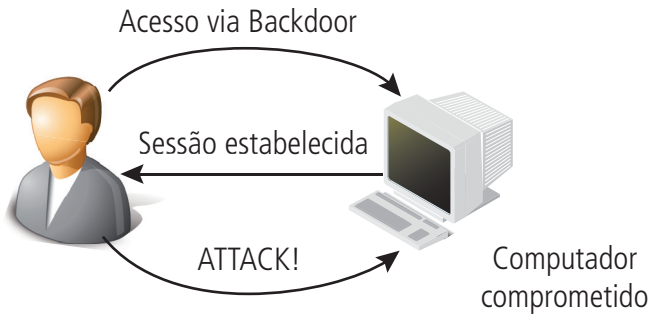


Figura 5.1: Exemplo de *backdoors*
Fonte: http://palizine.plynt.com/issues/2004Dec/application-backdoors/

5.3.5 *Buffer overflow*

Buffer overflow é uma anomalia ocorrida em um programa que, ao escrever dados em um *buffer*, ultrapassa os limites desse *buffer* e sobrescreve a memória adjacente. Estouros de *buffer* são disparados por entradas que são projetadas para executar código ou alterar o modo como o programa funciona. Isso pode resultar em comportamento errado do programa, incluindo erros de acesso à memória, resultados incorretos, parada total do sistema, ou uma brecha num sistema de segurança.

O *buffer overflow* utiliza-se de programas que não tratam a consistência dos dados de entrada, podendo haver uma desestruturação do código em execução, permitindo que um código estranho seja enviado e executado. Como exemplo, pode-se imagine um *buffer* de entrada de dados configurado para receber 32 *bytes*, o qual não possui uma checagem da consistência dos dados. O envio de um dado com mais de 32 *bytes* ocasionará o estouro do *buffer* (*buffer overflow*), e o restante do dado invadirá outras áreas de memória do sistema.

As formas mais comuns de *buffer overflow* são encontradas em servidores *web* e de FTP. Ao se submeter uma URL muito grande (geralmente acima de 150 caracteres) o servidor para de responder.

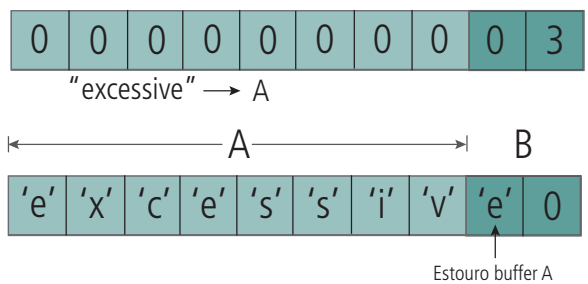


Figura 5.2: *Buffer overflow*
Fonte: https://commons.wikimedia.org/wiki/File:Buffer_overflow_basicexample.svg

5.3.6 *Password crackers*

Password crackers são, em sua grande maioria, programas de ataque de força bruta, que tentarão descobrir senhas através de combinações possível.

Os algoritmos de criptografia empregados para gerar senhas geralmente são públicos. Sua segurança reside em sua chave. Contudo, essa chave é pública. Assim sendo, o *password cracker* aplicará o algoritmo em cada combinação possível de letras até achar aquela que seja igual à senha criptografada original.

Geralmente *password crackers* são lentos, e sua eficiência depende inteiramente da qualidade das senhas. Senhas consideradas difíceis de quebrar, para um *passowrd cracker*, são aquelas que possuem letras, números, e caracteres de pontuação, como por exemplo: **! @#$%&*()[]{}-_=+<,>.?/**. Contudo, a melhor senha sempre será aquela sem sentido, randômica, e que use tais caracteres. Um típico *password cracker* levará algo em torno de dois a três anos de trabalho para quebrar uma senha de sete caracteres com essas características. Para cada novo caractere adicionado ao tamanho da senha, a dificuldade e o tempo sobem em ordem exponencial. Uma senha com 14 caracteres com tais características levaria milhares de anos. Com isso, chegamos à conclusão de que a senha ideal hoje possui pelo menos de 12 a 14 caracteres e as características descritas acima.

Além dos *password crackers* típicos, que usam a força bruta, existem aqueles que se baseiam em vulnerabilidades dos algoritmos de criptografia empregados. Estes não atacam por força bruta, mas fazendo o processo inverso dos algoritmos de criptografia, que são, geralmente baseados em algoritmos conhecidos, ou que tem o conhecimento de suas chaves.

5.3.7 *Exploits*

Um *exploit* é um programa de computador, uma porção de dados ou pequenos *scripts* que se aproveitam das vulnerabilidades de um sistema computacional ou serviços de interação de protocolos internet como, por exemplo, servidores *web*. Para um *exploit* atacar, o sistema precisa ter uma vulnerabilidade, ou seja, um meio de comunicação com a rede que possa ser usado para entrar. Geralmente são códigos locais (precisam ser executados no computador que se deseja comprometer), apesar de existirem *exploits* remotos (via rede).

O nome *exploit* também é atribuído às vulnerabilidades descobertas em *softwares* (sistemas operacionais, servidores, programas em geral). Existem diversos *sites* de segurança que falam sobre *exploits* mais recentes.

5.3.8 DoS (*Denial of Service*)

Um ataque de DoS, ou negação de serviço, é uma tentativa em tornar os recursos de um sistema indisponíveis para seus utilizadores. Alvos típicos são servidores *web* em que o ataque tenta tornar indisponíveis suas páginas. Não se trata de uma invasão do sistema, mas sim da sua invalidação por sobrecarga.

Os ataques do tipo DoS são usados muitas vezes em conjunto com invasões, ou porque alguns tipos de invasões exigem que determinados computadores não estejam funcionando (como no caso do *spoofing*) ou para despistar/desviar a atenção da invasão em si. Ataques DoS também são usados simplesmente para atrapalhar ou desacreditar um serviço. Os ataques DoS na sua grande maioria usam *buffer overflows* para conseguir obter sucesso.

5.3.9 DDoS (*Distributed Denial of Service*)

Os ataques do tipo DDoS (distribuído de negação de serviços) são uma evolução dos ataques de DoS e consistem geralmente em enviar para uma única máquina ou rede milhões de pacotes de rede ou requisições de serviço, em um dado momento. Obviamente, não existe maneira de gerar esse tráfego todo de um único ponto. Nesse tipo de ataque, um computador mestre tem sob seu comando milhares de computadores infectados e conhecidos como "zumbis".

A ideia do DDoS é fazer com que os zumbis, comandados pelo mestre, acessem, ao mesmo tempo, recursos de um mesmo servidor *web*. Como servidores *web* possuem um número limitado de usuários que eles pode atender simultaneamente, o grande e repentino número de requisições de acesso esgota esse número, fazendo com que o servidor não seja capaz de atender a mais nenhum pedido, levando-o a reiniciar ou até mesmo a ficar travado.

Existem outros tipos de pacotes ou requisições de conexão que têm uma eficácia muito maior do que uma simples requisição de acesso *web*. Contudo, o segredo está em como gerar este tráfego ou requisições, de várias máquinas espalhadas pela internet.

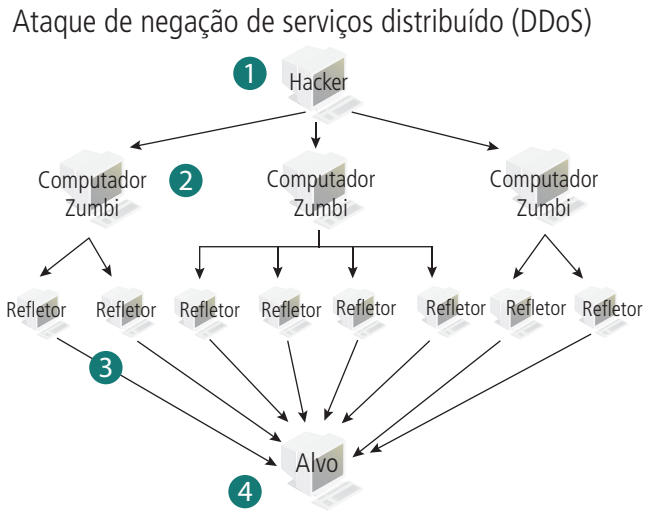


Figura 5.3: Ataque de DDoS
Fonte: em http://www.computacaoemalta.com/2011/06/ataque-de-negacao-de-servico-dos.html

5.3.10 IP *spoofing*

A técnica de *spoofing* é um ataque que consiste em mascarar (*spoof*) pacotes IP utilizando endereços de remetentes falsificados. Essa técnica é baseada na relação de confiança, existente na comunicação entre *hosts*, via protocolo IP. Como exemplo, imagine um determinado *host*, que será a vitima, e que só aceite comandos ou conexões de outro *host* que tenha um endereço IP pré-configurado. A técnica de *spoofing* consiste em que o invasor personifique esse *host* no qual a vítima confia.

Para a execução do ataque é necessário, basicamente, conhecer o endereço IP da vítima, o endereço IP do computador confiado, ter algum modo de tirar o computador confiado de operação, saber como quebrar o número de sequência TCP da vítima e assumir o lugar do computador confiado. Teoricamente, qualquer serviço que tenha sua segurança dependente apenas da confirmação de um endereço origem de rede é vulnerável a esse tipo de ataque.

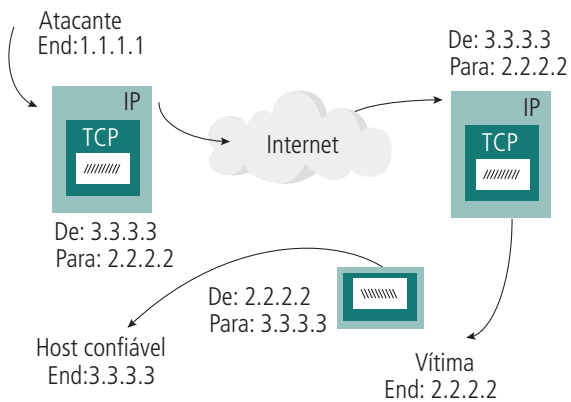


Figura 5.4: Ataque de IP *spoofing*
Fonte: http://pt.wikipedia.org/wiki/Ficheiro:IP_spoofing.png

5.3.11 Vírus

Um vírus de computador é um *software* malicioso que infecta os computadores, faz cópias de si mesmo e tenta se espalhar para outros computadores utilizando-se de diversos meios.

A maioria das contaminações ocorre pela ação do usuário ao executar o arquivo infectado recebido, como por exemplo, em um anexo de um *e-mail*. A contaminação também pode ocorrer por meio de arquivos infectados em mídias como, por exemplo, *pen drives*, quando eles são conectados aos computadores.

5.3.12 *Worms*

Um *worm* (verme, em português) é um programa autorreplicante, semelhante a um vírus. A diferença é que enquanto o vírus sempre precisa do programa hospedeiro que ele infectou para se propagar, o *worm* é um programa completo, que não precisa de outro programa para se autorreplicar.

5.4 Abordagem básica

Antes de partir para a tarefa de proteger um sistema, é necessário considerar os itens seguintes, segundo Nakamura (2007):

- identificar o que precisa ser protegido;
- determinar contra o que se quer proteção;
- determinar quais as possíveis ameaças;
- implementar as medidas para proteger o patrimônio com o melhor custo-benefício;
- rever o processo continuamente e promover melhorias a cada momento em que se descobre um ponto fraco.

5.5 Análise de risco

Na análise de risco há dois elementos a serem analisados: o patrimônio a ser protegido e as ameaças às quais esse patrimônio está sujeito.

5.5.1 Identificação do patrimônio

Normalmente, quando se fala do patrimônio de uma empresa em termos de informática, as primeiras referências são as informações valiosas e confidenciais e de propriedade intelectual. Porém, além desses itens, outros devem ser adicionados à lista do patrimônio a ser protegido em uma rede de computadores. Podemos citar:

- ***Hardware***: CPUs, placas, teclados, terminais, estações de trabalho, computadores pessoais, impressoras, *drives* de disco, linhas de comunicação, servidores de terminais e roteadores.

- ***Software***: programas-fonte, programas-objeto, utilitários, programas de diagnóstico, sistemas operacionais e programas de comunicação.

- **Dados**: em execução, armazenados na rede, arquivados fora da rede, *backups*, registros (*logs*) de auditoria, bancos de dados em trânsito pelo meio de comunicação.

- **Pessoas**: usuários, administradores e mantenedores de *hardware.*

- **Documentação**: em programas, *hardware*, sistemas, procedimentos administrativos locais.

- **Suprimentos**: papel, formulários, fitas, meios magnéticos.

5.5.2 Identificação das ameaças

As ameaças básicas ao patrimônio listado acima são as seguintes:

- acesso não autorizado a recursos e/ou informações;

- disponibilidade de conhecimento não intencional ou não autorizado de informações a terceiros;

- negação de serviço (sabotagem com objetivo de paralisar o serviço prestado).

5.6 Política de segurança

Políticas de segurança são sistemáticas gerenciais que visam determinar o nível de segurança de uma rede, sua funcionalidade e a facilidade de uso.

Essas sistemáticas são reunidas em um documento formal com regras às quais as pessoas deverão aderir para ter acesso à informação e à tecnologia de uma empresa. A política de segurança deve ser conhecida de todas as pessoas que usam o sistema e estão envolvidas nessa questão.

A política de segurança é um mecanismo preventivo de proteção dos dados e processos importantes de uma organização que define um padrão de segurança a ser seguido pelo corpo técnico e gerencial e pelos usuários, internos ou externos. Pode ser usada para definir as interfaces entre usuários, fornecedores e parceiros e para medir a qualidade e a segurança dos sistemas atuai. (LAUREANO, 2005).

Para se estabelecer uma política de segurança é necessário conhecer os objetivos, para depois poder medi-los. As políticas variam de organização para organização, pois, segundo Nakamura (2007), elas tratam dos aspectos humanos, culturais e tecnológicos de uma organização, levando também em consideração os processos e os negócios, além da legislação local. Baseados nisso, podemos considerar os seguintes pontos ao se estabelecer uma política de segurança:

- **Serviço oferecido x segurança proporcionada**: todo serviço oferecido representa um risco a mais para a segurança. É necessário considerar se a perda proporcionada pelo risco da segurança vai compensar os ganhos com o serviço oferecido.

- **Facilidade de uso x segurança**: toda prática de segurança impõe uma dificuldade de uso para os usuários. Deve-se considerar se o esquema de segurança que se pretende adotar não impõe uma carga excessiva de dificuldade para o usuário.

- **Custo da segurança x risco de perda**: não ter segurança nenhuma evolve um risco grande de perda. A segurança máxima imprime um custo para mantê-la. Há que se encontrar um ponto ótimo, em que o custo da segurança compense as possíveis perdas.

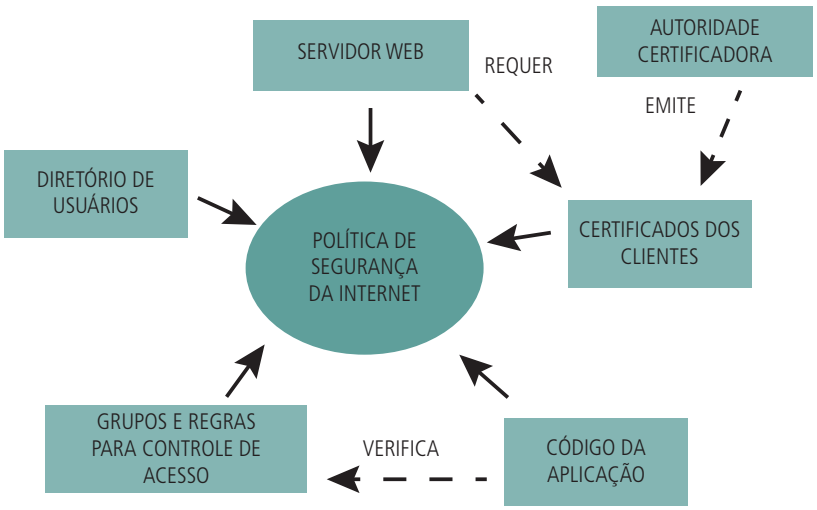


Figura 5.5: Política de segurança
Fonte: http://www.rnp.br/newsgen/0011/ldap.html

5.6.1 Função da política de segurança

De acordo com Laureano (2005), a política de segurança deve ir além dos aspectos relacionados com sistemas de informação ou recursos computacionais; ela deve estar integrada com as políticas institucionais, metas de negócio e com o planejamento estratégico da empresa. Baseada nesse panorama, a política de segurança deve:

- informar os usuários, funcionários e gerentes das exigências obrigatórias para proteger seu patrimônio tecnológico e de informações;
- especificar os mecanismos através dos quais essas exigências podem ser cumpridas;
- emitir diretrizes para aquisição, configuração e auditoria de sistemas e redes de computador para atender à política;
- explicitar o que os usuários podem e não podem fazer nos vários componentes do sistema, incluindo o tipo de tráfego permitido nas redes e quais são esses usuários;
- evitar ambiguidades e mal-entendidos.

5.6.2 Quem deve se envolver com segurança

De acordo com Laureano (2005), é necessário descrever as responsabilidades (e, em alguns casos, os privilégios) de cada classe de usuários do sistema. Assim, devemos definir os envolvidos com a segurança da informação e suas funções. Os principais elementos envolvidos com a segurança são:

- Administrador de segurança do *site*;
- Pessoal de informática (ex., do CPD);
- Administradores de grupos grandes de usuários (ex., divisão de negócios, departamento de ciência da computação de uma universidade, etc.);
- Brigada de segurança (equipe de emergência acionada em caso de invasão);
- Representantes dos grupos de usuários afetados pela política de segurança;
- Gerência responsável;
- Departamento jurídico (se apropriado);
- Auditoria.

5.6.3 Fatores de sucesso de uma política de segurança

O planejamento é o fator crítico de sucesso para a segurança da informação. Alguns pontos devem ser considerados para o sucesso da política:

- Ser implementável pelos procedimentos administrativos de sistema;
- Ser aplicável e sancionável pelas ferramentas de segurança;
- Definir responsabilidades dos usuários, administradores e gerência.

5.6.4 Componentes de uma política de segurança

Segundo Nakamura (2007), uma política de segurança deve conter:

- Diretrizes de aquisição de informática;
- Política de privacidade, tal como monitoração de correio eletrônico, registro de digitação e acesso a arquivos de usuários;
- Definição dos direitos e dos privilégios de acesso;
- Política de responsabilidades para os usuários, equipe operacional e gerência; deve especificar métodos de auditoria;
- Política de autenticação;

- Esquema de disponibilidade do sistema;
- Política de manutenção da rede e dos sistemas (ex., permitir ou não acesso remoto e definir seu controle);
- Política de registro e aviso de violações;
- Informações de suporte;
- Exigências regulamentares (ex., monitoração de linha).

5.6.5 Flexibilidade da política de segurança

Toda regra tem exceções. Portanto, se possível, definir em que casos essas regras podem apresentar exceções.

5.6.6 Plano global de segurança

De acordo com Nakamura (2007), a política de segurança deve fazer parte de um plano abrangente de segurança e ser consistente com ele.

O plano de segurança deve definir a lista dos serviços de rede oferecidos, quais áreas da organização oferecerão tais serviços, quem terá acesso a eles, como esse acesso será feito, quem vai administrar esses serviços. Esse plano deve definir as classes de incidentes e as respostas correspondentes. Também deve estar preocupado não só com a segurança interna, mas também com a externa.

5.7 Modelos de segurança

Ao construirmos uma modelo de segurança, podemos definir que existem dois modelos extremos, que são a negação total de acessos e a permissão total. Um bom modelo deve estar a meio termo, lembrando que quanto maior o nível de segurança mais lento é o sistema.

5.7.1 Negação total

O modelo de negação total é aquele em que, a princípio, todos os serviços possíveis são negados e que todas as restrições possíveis são aplicadas. À medida que houver necessidade de liberação de serviços e concessão de permissão, vai-se concedendo. É um modelo mais eficiente em termos de segurança, porém mais difícil de implementar, já que todas as brechas possíveis têm que ser identificadas e fechadas. Nesse modelo a premissa básica é: "**o que não é permitido é proibido**".

5.7.2 Permissão total

O modelo de permissão total é exatamente o oposto. Um mínimo de segurança é aplicado. À medida que vão surgindo problemas relativos à segurança, vão sendo implementadas restrições. É um modelo mais fácil de implementar, porém mais fraco.

Os dois modelos não são mutuamente exclusivos para um *site*. É possível ter alguns servidores com um modelo e outros com o outro modelo, dependendo do nível de segurança que se queira aplicar para cada um deles. Cuidado deve ser tomado, porém, ao dosar os dois modelos em redes corporativas. Isso para evitar a aplicação do mesmo modelo para necessidades de segurança diferentes e, ao mesmo tempo, evitar que servidores com um tipo de modelo não comprometam a segurança e a operação de servidores com outro tipo.

5.8 Segurança dos serviços

É prudente e recomendável que serviços diferentes, alvos de níveis de segurança diferentes, estejam devidamente separados em uma rede. É bastante comum que as organizações ofereçam, a partir de suas páginas *web*, a transferência (*download*) de arquivos, programas, informações de catálogos, manuais, artigos, etc. Normalmente essa transferência é anônima. Se o serviço de transferência está instalado na mesma máquina do servidor HTTP (*web*), é provável que o intruso venha a ter acesso aos arquivos HTML que compõem as páginas, podendo desfigurá-las. Por isso é recomendável, portanto, separar as aplicações conforme sua categoria em relação à segurança.

Também é sempre bom frisar que a parte mais vulnerável de uma rede é que define todo o seu nível de segurança, por mais que haja outros pontos mais fortificados.

A regra geral é rodar cada tipo de serviço, tanto quanto possível, em uma máquina diferente, com níveis de segurança definidos para cada máquina.

5.8.1 Identificação da real necessidade dos serviços

É importante definir quais serviços realmente são ou serão necessários. A complexidade de um sistema de segurança pode crescer exponencialmente com o número de serviços prestados. Alguns serviços, tais como transferência de arquivo de acesso anônimo, servidores www, etc., podem expor a rede a brechas de segurança. É necessário avaliar onde esses serviços serão instalados e se isso valerá a pena.

5.9 Mecanismos de segurança

Segundo Nakamura (2007), os mecanismos de segurança podem ser divididos baseados nos controles físicos e lógicos:

- **Controles físicos**: são barreiras que limitam o contato ou acesso direto à informação ou à infraestrutura, além de prover proteção contra danos provocados pela natureza. Assim, mecanismos de segurança que apoiam os controles físicos como portas, trancas, paredes, blindagem, guardas, etc., devem ser usados. Também não se deve esquecer a proteção contra incêndios, inundações, falta de energia, ou contra os agentes da natureza.

- **Controles lógicos**: são barreiras que impedem ou limitam o acesso à informação que está em ambiente controlado, geralmente eletrônico, e que, de outro modo, ficaria exposta à alteração não autorizada por elemento mal-intencionado.

5.10 Proteção da infraestrutura

Os componentes físicos e lógicos dos servidores e das redes são alvos de ataques visando comprometer seu desempenho ou negar seus serviços. Os ataques mais comuns a servidores passam por explorar possíveis portas de serviço que estão ativadas e passar pela barreira das senhas.

Já as redes podem sofrer inundação de pacotes, cujo objetivo é degradar seu desempenho ou até chegar ao ponto de fazer com que elas não possam mais transmitir dados úteis. A inundação mais eficiente está na escolha do pacote certo que possa causar sua multiplicação. Não há maneiras de se evitar o ataque de inundação de pacotes, mas esse ataque é facilmente detectável e relativamente simples de ser terminado.

5.11 Proteção de serviços

Para Nakamura (2007), ao se desenvolver uma política deve-se considerar três regiões principais para as quais uma atenção diferente deve ser dispensada. O ambiente interno, o ambiente externo, e um terceiro, que não é nem interno nem externo, para o qual uma estrutura totalmente separada deve ser construída.

De acordo com Laureano (2005), vários são os serviços mais comuns, os quais estão sujeitos a sofrer mais ataques na rede, que são os serviços de

resolução de nomes (DNS), os serviços de senhas/chaves, os serviços de autenticação/proxy, correio eletrônico, www, transferência de arquivo e NFS.

5.11.1 DNS (*Domain Name Servers*)

O serviço DNS não possui intrinsecamente nenhum tipo de segurança e pode ser usado por um intruso para controlar o seu servidor ou se fazer passar por um, de modo a desviar o tráfego da rede para fins de quebra de segurança. O tráfego pode ser desviado para ser monitorado, ou usuários podem ser enganados a ponto de revelar suas senhas de autenticação.

Uma proteção para esse tipo de ataque é a criação de DNS secundário em redes protegidas e DNS primário protegido da paralisação de serviço (*Denial of Service*) por roteadores com filtros.

5.11.2 Servidores de senhas/chaves

Servidores de senhas e chaves devem estar protegidos por algoritmos de criptografia. Mas as senhas podem ser desvendadas por *softwares* que testam palavras do dicionário, criptografando-as de modo a verificar se alguma delas é igual a uma das senhas criptografadas.

A melhor forma de proteção é habilitar estritamente os servidores que terão acesso ao servidor de senhas; mesmo assim, é preciso limitar os serviços para senhas, desativando os de Telnet e FTP.

5.11.3 Servidores de *proxy* e autenticação

Como os servidores de *proxy* e autenticação afunilam o tráfego de acesso, acabam sendo alvo de ataques.

A regra é limitar o acesso somente a servidores autorizados e restringir os serviços somente àqueles que serão efetivamente usados.

5.11.4 Correio eletrônico

Os protocolos de correio eletrônico são os mais antigos e mais largamente usados. Por isso, são alvos constantes de ataques. Além disso, o servidor de correio eletrônico, por natureza, se conecta e aceita entradas do mundo exterior. Outro fator de complicação é que o servidor de correio entrega mensagens para todos os usuários, mas seu conteúdo é privado. Assim, ele acaba tendo privilégios de acesso à raiz do sistema, o que abre uma série de brechas de segurança.

Como os servidores de correio eletrônico normalmente são compostos de dois agentes, um que recebe e envia mensagens e outro que estabelece o processamento, a solução está na separação dos agentes. A instalação desse tipo de estratégia requer cuidado para não deixar falhas de segurança.

5.11.5 WWW (*world wide web*)

Os servidores www têm uma natureza eminentemente pública e são o principal alvo de ataques. É importante que todos eles estejam em computadores totalmente separados das aplicações críticas de uma empresa. Alguns *sites* incorporam serviços de FTP (transferência de arquivos) em seu serviço www. O acesso a esses servidores é normalmente anônimo (não requer identificação do usuário) e assim são os serviços FTP dessas páginas.

Nos dois casos é importante que o servidor seja habilitado apenas para leitura; caso contrário, o invasor poderá alterar o conteúdo do servidor www.

5.11.6 FTP (*File Transfer Protocol*)

Servidores FTP configurados inadequadamente podem permitir que intrusos copiem, substituam ou apaguem arquivos à vontade em qualquer lugar do servidor. Acesso a senhas criptografadas, dados proprietários e a introdução de *trojan* ("cavalos de Troia") são algumas das brechas potenciais de segurança que podem ocorrer quando o serviço é configurado incorretamente.

Servidores FTP devem ser disponibilizados em computadores apenas para esse fim, especialmente os que aceitam escrita de arquivos. Não se recomenda colocar servidores FTP nos mesmos computadores usados pelos servidores www.

O protocolo TFTP (Trivial FTP) também não é recomendado para uso generalizado, pois não requer autenticação do usuário. Esse protocolo é muito usado para carga de arquivo de configuração de roteadores. Esse serviço deve rodar em seu próprio servidor.

5.11.7 NFS

O NFS (*Network File Service*) permite que os servidores compartilhem discos comuns. O NFS não tem nenhum esquema de segurança e, portanto, deve ser liberado somente para os servidores que estarão usando efetivamente o serviço. A proteção contra acesso externo ao NFS deve ser feita por um *firewall*.

5.11.8 Telnet

Telnet é um serviço de conexão remota via internet. Ele permite que um computador trabalhe como usuário de outro. Essa é uma das brechas preferidas dos invasores, já que se pode controlar um computador remotamente.

Muitos administradores de rede usam esse serviço para manutenção remota. Porém não é o tipo de serviço que deva ser oferecido ao público em geral; se for, deve estar em um ambiente separado do resto da rede para evitar ataques. Em geral, se não for usado, deve ser desativado.

5.11.9 Proteção da proteção

Assista ao vídeo "Dicas de segurança da Empresa de Correios e Telégrafos (ECT) disponível em http://www.youtube.com/watch?v=xIPgmCGX7i4

Não se deve esquecer que os servidores e roteadores que estarão proporcionando segurança deverão também ser protegidos contra ataques. Eles não devem ser acessíveis externamente; devem oferecer acesso mínimo para usuários do *site*, exceto para autenticação. Também devem ser separados dos outros servidores. Todo acesso ao servidor ou roteador de proteção deve ser registrado, de modo a ser possível rastrear os eventos em caso de quebra da segurança.

5.12 Procedimentos de segurança

De acordo com Laureano (2005), vários são os procedimentos de segurança que devem ser seguidos após a implementação da política, pois, caso contrário, de nada adiantará a existência da política.

5.12.1 Acesso lógico

São os acessos aos sistemas e programas de uma rede. É necessário exigir que usuários se identifiquem ao acessar os recursos computacionais de uma rede. Existem mecanismos de segurança que apoiam os controles lógicos, como por exemplo:

- **Mecanismos de criptografia**: permitem a transformação reversível da informação de forma a torná-la ininteligível a terceiros. Utilizam-se, para tal, algoritmos determinados e uma chave secreta para, a partir de um conjunto de dados não criptografados, produzir uma sequência de dados criptografados. A operação inversa é a decifração.

- **Assinatura digital**: é um conjunto de dados criptografados, associados a um documento do qual são função, garantindo a integridade do documento associado, mas não a sua confidencialidade.

- **Mecanismos de garantia da integridade da informação**: usam funções de *hashing* ou de checagem, consistindo na adição.

- **Mecanismos de controle de acesso**: utilizam palavras-chave, sistemas biométricos, *firewalls*, cartões inteligentes.

- **Mecanismos de certificação**: atestam a validade de um documento.

- **Integridade**: medida em que um serviço/informação é genuíno, isto é, está protegido contra a personificação por intrusos.

- ***Honeypot***: é o nome dado a um *software* cuja função é detectar ou impedir a ação de um *cracker*, de um *spammer*, ou de qualquer agente externo estranho ao sistema, enganando-o, fazendo-o pensar que esteja de fato explorando uma vulnerabilidade daquele sistema.

5.12.2 Autenticação

Por muitos anos apenas a senha e o nome do usuário eram usados para autenticá-lo. Como o uso de redes era limitado, as senhas eram passadas em texto puro, porque o risco era mínimo. Hoje, as redes permeiam as empresas ao redor do globo. Com o advento de novas tecnologias, as senhas são criptografadas com chaves não reutilizáveis e autenticadas com uso de cartões inteligentes.

Figura 5.6: Exemplo de autenticação de usuário
Fonte: http://www.seuenium.com.br/tag/selenium/

Medidas que devem ser tomadas para proteger senhas:

- **Estabelecer senhas robustas**: elas devem ser robustas contra ataque de força bruta. Ou seja, não devem formar palavras em nenhuma língua, nem siglas comuns, comerciais ou culturais e devem ser longas e consistir de uma mistura de letras maiúsculas, minúsculas, números e caracteres

especiais. À medida que a tecnologia evolui, o tamanho mínimo da senha continua a crescer. Faça com que a técnica de senhas e a criptografia acompanhem os últimos avanços.

- **Alterar senhas *default***: muitos sistemas operacionais e programas aplicativos são instalados com senhas e contas *default*. Elas devem ser mudadas imediatamente para que não possam ser adivinhadas ou quebradas.

- **Restringir o acesso ao arquivo de senhas**: uma técnica eficiente é usar um arquivo de senhas legítimas bem protegido em alguma parte do sistema e colocar outro arquivo com senhas falsas ou inativas no arquivo padrão de senhas. Esta técnica é conhecida como *shadow password.*

- **Trocar senhas**: há controvérsias sobre se deve forçar usuários a trocar senhas ruins por boas e quando fazer isso. No entanto, uma prática razoável seria trocar a senha uma vez por ano ou estimar quanto tempo seria necessário, dado o poder computacional atual, para que um *hacker* pudesse quebrar a senha. Outra prática absolutamente necessária é trocar as senhas sempre que uma senha privilegiada é violada. Por exemplo, se a senha de um administrador é violada, todas as senhas do sistema devem ser trocadas.

- **Bloqueio de senhas**: muitos sistemas de autenticação bloqueiam o acesso do usuário sempre que houver mais de três tentativas (sem sucesso) consecutivas de entrar no sistema. Alguns sistemas bloqueiam permanentemente, forçando os usuários a contatarem o administrador; outros sistemas bloqueiam por um tempo determinado, variando de minutos, horas ou até dias.

5.12.3 Senha única

Com o uso de "cavalos de Troia" e *sniffers*, *hackers* podem detectar senhas que passam em claro pela rede. Para resolver esse problema, vários mecanismos foram desenvolvidos, tais como a técnica de resposta a desafio que usa uma senha diferente por vez.

5.12.4 *Firewalls*

Um *firewall* é um dos muitos mecanismos usados para controlar e vigiar o acesso à rede. Mas ele é uma medida a mais que deve ser cuidadosamente estudada e usada em conjunto com uma série de medidas de segurança.

O trabalho de um *firewall* é permitir ou não a entrada ou a saída de um pacote de uma rede interna para o mundo exterior, baseado em certas características do pacote, tais como endereço de origem e destino, porta de serviço, tipo de serviço, entre outros.

A terminologia e a teoria sobre *firewalls* variam bastante. Os *firewalls* nem sempre são uma única máquina. Às vezes, os *firewalls* podem ser uma combinação de roteadores, segmentos de rede, computadores, etc. É importante notar que a configuração de um *firewall*, qualquer que seja ele, requer grande conhecimento de TCP/IP.

Os *firewalls*, como qualquer outro sistema de segurança, devem estar em constante atualização. Assim, eles sempre requerem manutenção, instalação de *patches* e atualizações e monitoração regulares.

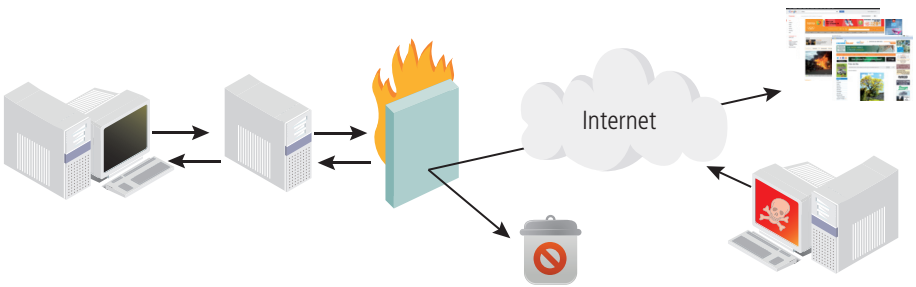


Figura 5.7: Exemplo de *firewall*
Fonte: http://www.tecportal.com.br/configuracoes-de-seguranca-para-o-firewall/

5.12.5 *Backups*

Do ponto de vista da segurança:

1. Garanta a execução de *backups* periódicos.

2. Armazene o backup em outra localidade.

3. Use criptografia se possível, mas certifique-se de que o *software* para decifrar esteja disponível em qualquer época em que seja necessário recuperar o *backup*. Também mantenha um esquema cuidadoso de chaves e/ou senhas;

4. Nunca assuma que o *backup* está correto. Muitos ataques ocorrem bem antes do incidente notado. Faça verificações periódicas do *backup* para descobrir se não carrega, de alguma forma, uma invasão.

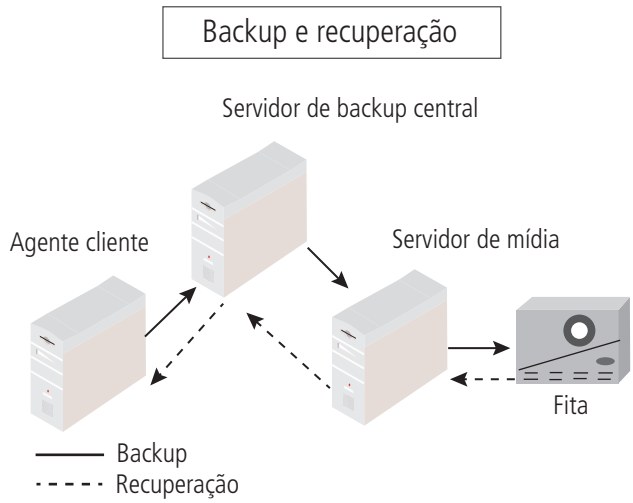


Figura 5.8: *Backup*
Fonte: http://technet.microsoft.com/pt-br/library/bb821259.aspx

5.12.6 Acesso público

Acesso público são áreas projetadas para dar comodidade a usuários se conectarem com sua rede a distância, quer seja por um terminal estabelecido por sua empresa (como um terminal de caixa eletrônico ou *cyber* cafés), quer seja por um ponto de rede para conexão de equipamentos portáteis. Essa modalidade de acesso permite o uso de IPs falsos, rastreamento de pacotes, etc. Se for essencial prestar esse tipo de serviço ao público, estabeleça-o numa rede separada da rede interna de sua empresa. Não só isso, mas deve-se vigiar escritórios vazios e é uma boa ideia desconectá-los fisicamente do quadro de distribuição de cabos e monitorar toda a tentativa de conexão não autorizada.

5.12.7 Redes de dados públicas

As redes que são oferecidas por operadoras de serviços públicos de telecomunicações devem ter a devida atenção no que tange à segurança, já que os invasores têm tanto interesse nessas redes como nas redes internas das empresas. É recomendável que o uso dessas redes envolva criptografia e circuitos dedicados.

5.13 Criptografia

A proteção de dados sigilosos é uma necessidade antiga. Com o advento da internet e de sua consequente facilidade de transmitir dados de maneira precisa e extremamente rápida, a criptografia tornou-se uma ferramenta fundamental para permitir que apenas o emissor e o receptor tenham acesso livre à informação trabalhada.

O termo criptografia surgiu da fusão das palavras gregas *kryptós* e *gráphein*, que significam oculto e escrever, respectivamente. De acordo com Kurose (2010), as técnicas criptográficas permitem que um remetente disfarce os dados de modo que um intruso não consiga obter nenhuma informação dos dados interceptados.

5.13.1 Técnicas criptográficas

Para Nakamura (2009), a cifragem (*encrytion*) é o processo de disfarçar o texto claro (*plaintext* ou *cleartext*), de tal modo que sua substância é escondida em um texto cifrado (*ciphertext*), enquanto que a decifragem (*decryption*) é o processo de transformar o texto cifrado de volta em texto claro original. Os processos de cifragem e decifragem são realizados com o uso de algoritmos com funções matemáticas.

As técnicas criptográficas mais conhecidas envolvem o conceito de chaves, as chamadas chaves criptográficas. Trata-se de um conjunto de *bits* baseado em um determinado algoritmo capaz de codificar e de decodificar informações. Se o receptor da mensagem usar uma chave incompatível com a chave do emissor, não conseguirá extrair a informação.

Figura 5.9: Exemplo de criptografia
Fonte: http://www.infowester.com/criptografia.php

As chaves de criptografia possuem tamanhos múltiplos de 64 *bits*, como por exemplo, chave de 128 *bits* e 512 *bits*. Quanto mais *bits* forem utilizados, mais segura será a criptografia. Isso se deve ao fato de estarmos trabalhando com *bits*, ou seja, a base binária. Então, se um algoritmo use chaves de oito *bits*, ele permite que apenas 256 (28) chaves possam ser usadas na decodificação. Isso deixa claro que o uso de oito *bits* é inseguro, pois gerar as 256 combinações é muito simples para os computadores atuais, devido à grande capacidade de processamento. Se forem usados 128 ou mais *bits* para chaves, o número de combinações aumenta sensivelmente (faça 2 elevado a 128 para ver o que acontece), deixando a informação criptografada bem mais segura.

5.13.2 Criptografia de chave simétrica

A criptografia de chave privada ou simétrica é responsável pelo sigilo das informações por meio de utilização de uma chave secreta para a codificação e decodificação dos dados (NAKAMURA, 2009).

Exemplos de vários algoritmos que usam chaves simétricas:

- **DES (*Data Encryption Standard*)**: criado pela IBM em 1977, faz uso de chaves de 56 *bits*. Isso corresponde a 72 quatrilhões de combinações. É um valor absurdamente alto, mas não para um computador potente. Em 1997, esse algoritmo foi quebrado por técnicas de "força bruta" (tentativa e erro) em um desafio promovido na internet.

- **IDEA (*International Data Encryption Algorithm*)**: criado em 1991 por James Massey e Xuejia Lai, o IDEA é um algoritmo que faz uso de chaves de 128 *bits* e que tem uma estrutura semelhante ao DES. Sua implementação em *software* é mais fácil do que a implementação deste último.

- **RC (*Ron's Code ou Rivest Cipher*)**: criado por Ron Rivest na empresa RSA Data Security, esse algoritmo é muito utilizado em *e-mails* e faz uso de chaves que vão de oito a 1.024 *bits*. Possui várias versões: RC2, RC4, RC5 e RC6. Essencialmente, cada versão difere da outra por trabalhar com chaves maiores.

- **Outros**: há ainda outros algoritmos conhecidos, como o AES (*Advanced Encryption Standard* – que é baseado no DES), o 3DES, o *Twofish* e sua variante *Blowfish*, entre outros.

Ainda de acordo com Nakamura (2009), o uso de chaves simétricas tem como característica a rapidez na execução, porém existe o problema da necessidade de distribuição da chave secreta, que deve ser feita de forma segura, pois tanto o emissor quanto o receptor precisam conhecer a mesma chave. Assim, essa cifragem não é adequada em situações nas quais a informação é muito valiosa, pois é necessário usar uma grande quantidade de chaves para o caso em que muitas pessoas ou entidades estejam envolvidas, além do que a transmissão dessa chave de um para o outro pode tornar o sistema vulnerável.

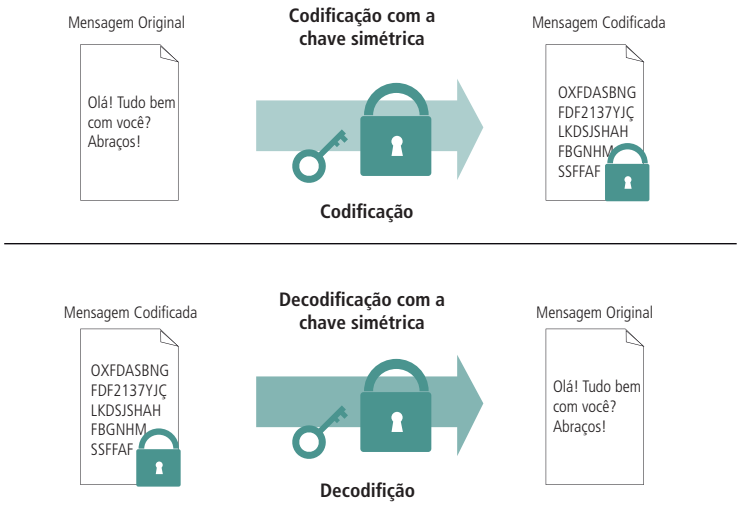


Figura 5.10: Criptografia simétrica
Fonte: http://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2008_2/hugo/index.html

5.13.3 Criptografia de chave assimétrica

De acordo com Nakamura (2009), os algoritmos de chave pública ou assimétrica podem possibilitar sigilo, integridade, não repúdio e autenticidade, pois trabalham com um par de chaves diferentes, uma denominada privada e outra denominada pública.

Nesse método, um emissor deve criar uma chave de codificação e enviá-la ao receptor. Essa é a chave pública. Outra chave deve ser criada para a decodificação. Esta, a chave privada, é secreta. Ou seja, usa-se uma chave para criptografar a mensagem e outra para descriptografar. E o sistema só trabalha com esse par: chave de criptografia/chave de descriptografia. A mensagem que foi criptografada com a respectiva chave do par só pode ser descriptografada com a respectiva chave do par. A chave para criptografar é pública, e a chave de descriptografia só é conhecida pelo destinatário.

Explica-se: ao enviar uma mensagem, procura-se o par chave de criptografia/chave de descriptografia pertencente ao destinatário e encriptografa a mensagem com essa chave pública do par. Quem conhece a chave de descriptografia do par é só o destinatário; então, apenas ele terá acesso à informação criptografada com sua chave pública.

O algoritmo assimétrico minimiza o problema de troca de chaves, pois não é necessário um canal seguro para tal. Porém, ele é cerca de 60 a 70 vezes mais lento que os algoritmos simétricos (NAKAMURA, 2009).

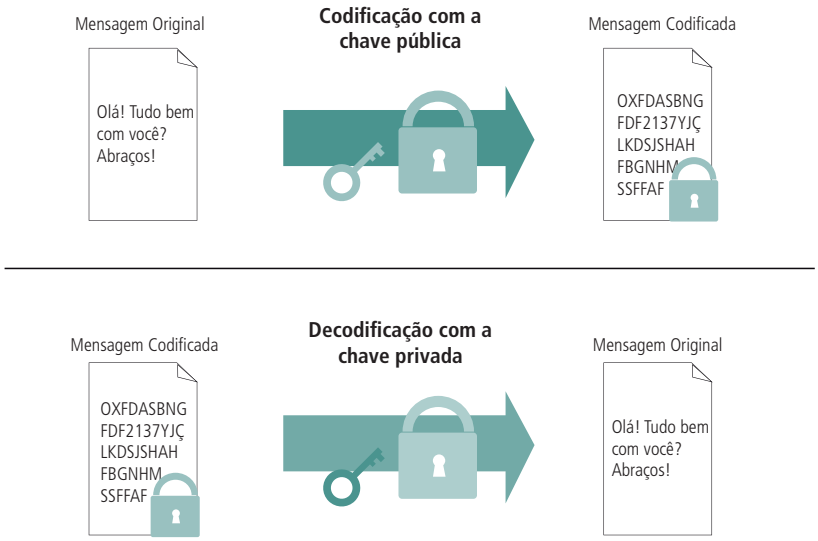


Figura 5.11: Exemplo de criptografia assimétrica
Fonte: http://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2008_2/hugo/index.html

Exemplos de algoritmos que usam chaves assimétricas:

- **RSA (*Rivest, Shamir and Adleman*)**: criado em 1977 por Ron Rivest, Adi Shamir e Len Adleman nos laboratórios do Massachusetts Institute of Technology (MIT), é um dos algoritmos de chave assimétrica mais usado. Nele, números primos (número primo é aquele que só pode ser dividido por 1 e por ele mesmo) são utilizados da seguinte forma: dois números primos são multiplicados para se obter um terceiro valor. Porém, descobrir os dois primeiros números a partir do terceiro (ou seja, fazer uma fatoração) é muito trabalhoso. Se dois números primos grandes (realmente grandes) forem usados na multiplicação, será necessário usar muito processamento para descobri-los, tornando essa tarefa praticamente inviável. Basicamente, a chave privada no RSA são os números multiplicados e a chave pública é o valor obtido.

- **ElGamal**: criado por Taher El Gamal, esse algoritmo faz uso de um problema matemático conhecido por "logaritmo discreto" para se tornar seguro. Sua utilização é frequente em assinaturas digitais.

- **Outros:** existem ainda outros algoritmos, como o DSA (*Digital Signature Algorithm*), o Schnorr (praticamente usado apenas em assinaturas digitais) e *diffie-hellman*.

5.13.4 Criptografia híbrida

Como já foi dito, a criptografia assimétrica, apesar de ser mais segura que a simétrica, é mais lenta para ser processada. Para equacionar o problema velocidade *versus* segurança, foi desenvolvido o sistema de criptografia híbrida, que consegue unir a segurança da criptografia assimétrica com a rapidez de processamento da simétrica. Na criptografia híbrida, a mensagem original é primeiramente criptografada como a chave pública do destinatário, formando uma nova mensagem (mensagem1), Em seguida a mensagem1 é criptografada com uma chave simétrica formando outra nova mensagem (mensagem2). A mensagem2 é enviada ao destinatário, junto com a chave simétrica. Ao chegar ao destinatário, ele descriptografa a mensagem2 com a chave simétrica enviada conjuntamente com a mensagem, tendo acesso à mensagem1. Então, ele descriptografa a mensagem1 com a sua chave privada, tendo acesso à mensagem original.

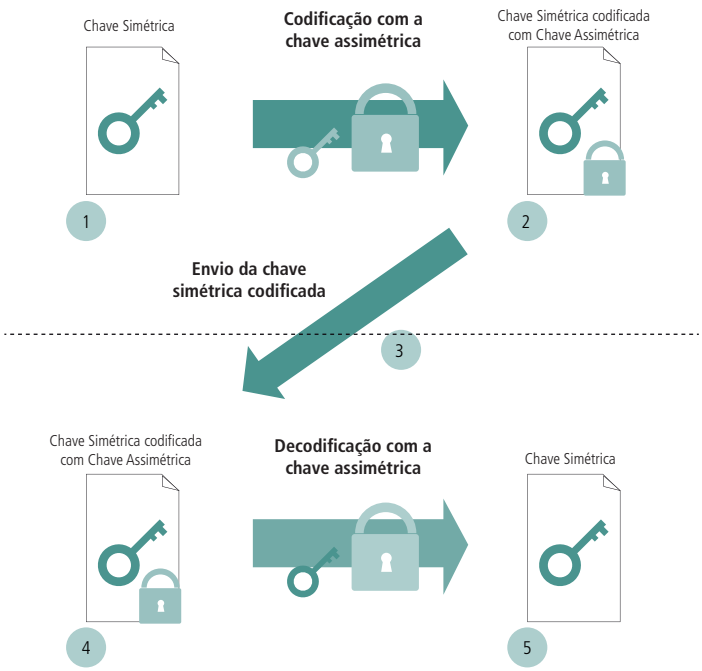


Figura 5.12: Criptografia híbrida
Fonte: http://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2008_2/hugo/index.html

5.14 Certificação digital

De acordo com Kurose (2010), a certificação digital, ou assinatura digital, é um método criptográfico usado para demonstrar quem é o dono ou criador de um documento ou deixar claro que alguém concorda com o conteúdo de um documento. Ela nada mais é que uma autenticação de informa-

Para saber mais sobre criptografia acesse http://w3.ualg.pt/~hshah/algoritmos/aula20/aula20.htm

ção digital. Normalmente a assinatura digital vai junto com um documento eletrônico em meio digital, tal qual uma assinatura em um documento em papel, e dessa maneira permite provar que o documento foi mesmo emitido por quem enviou o arquivo eletrônico.

Para Nakamura (2009), o destinatário da mensagem eletrônica recebe-a assinada com a chave privada do remetente e usa a chave pública correspondente desse destinatário para verificar a assinatura digital; ou seja, a certificação digital funciona de maneira contrária à criptografia assimétrica.

Os sistemas de assinatura digital mais conhecidos são:

- **PGP:** é a sigla de *Pretty Good Privacy* que oferece *e-mail* seguro (KUROSE, 2010). Por ser disponibilizado de forma gratuita, acabou se tornando uns dos meios de criptografia mais conhecidos na troca de *e-mails*. Para reforçar a segurança, o *software* pode realizar um segundo tipo de criptografia através de um método conhecido como "chave de sessão" que, na verdade, é um tipo de chave simétrica.

- ***Kerberos*:** opera com uma base de dados de chave simétrica usando um centro de distribuição de chaves conhecido como servidor *Kerberos*. A um usuário ou serviço são concedidos tíquetes depois de se comunicar apropriadamente com o servidor *Kerberos*. Esses tíquetes são usados para autenticação entre os usuários ou serviços. Todos os tíquetes incluem uma marcação de tempo que limita o período para o qual são válidos. Dessa forma, clientes e servidores devem ter uma fonte de tempo segura e devem ser capazes de manter a precisão do tempo.

5.15 *Hash*

Qualquer participante pode verificar a autenticidade de uma assinatura digital, bastando decifrá-la com a chave pública do signatário, à qual todos podem ter acesso. Se o resultado é significativo, está garantido o uso da chave secreta correspondente na assinatura e, portanto, sua autenticidade. Resta ainda comprovar a associação da assinatura ao documento, o que é feito recalculando o *hash* do documento recebido e comparando-o com o valor incluído na assinatura. Se forem iguais, prova-se ainda a ligação com o documento, assim como a sua integridade (não alteração). Uma vez que a verificação é realizada utilizando a chave pública, sua validação pode ser realizada por terceiros, tais como árbitros e auditores (LAUREANO, 2005).

Por isso o *hash* é considerado uma espécie de um lacre digital que representa o conteúdo de um arquivo de dados Os lacres produzidos são de comprimento fixo, não importando o comprimento do arquivo que eles representem. Qualquer alteração efetuada no arquivo, por mínima que seja, altera substancialmente o resultado *hash*, indicando que o arquivo foi alterado. Os algoritmos de *hash* mais usados são os de 16 *bytes* MD4 e MD5 ou o SHA-1, de 20 *bytes*.

Os *hashes* são muito usados para autenticação da integridade de arquivos e validação de senhas e assinaturas digitais. O exemplo da figura 5.13 mostra a função *hash* gerada pelo algoritmo MD5. Note que a alteração da primeira letra do fluxo da palavra "Aldeia NumaBoa" de A maiúsculo para a minúsculo, gerou funções *hash* diferentes.

```
Aldeia NumaBoa      3cdb658425ee484e4bff3d4583f6f851
aldeia NumaBoa      9c1f41ef263026b0283676d63df21fd1
```

Figura 5.13: Exemplo de *hash*
Fonte: http://www.numaboa.com.br/index2.php?option=com_content&do_pdf=1&id=340

Figura 5.14: Fluxo de envio de um documento eletrônico de forma segura
Fonte: Elaborada pelo autor

5.16 Auditoria

A auditoria é a rastreabilidade dos diversos passos que um negócio ou processo realizou ou a que uma informação foi submetida, identificando os participantes, os locais e horários de cada etapa. Auditoria em *software* significa uma parte da aplicação, ou conjunto de funções do sistema, que viabiliza uma auditoria; consiste no exame do histórico dos eventos dentro de um sistema para determinar quando e onde ocorreu uma violação de segurança (LAUREANO, 2005). Auditoria envolve a coleta de dados gerados pela atividade da rede, a qual pode ser útil na análise de eventos para responder a incidentes de segurança.

5.16.1 Coletando os dados a serem auditados

Dados para auditoria devem incluir qualquer tentativa de acessar um nível de segurança diferente por qualquer pessoa, processo ou outra entidade da

rede. Isso inclui *login*, *logout*, acesso de "superusuário" ou "admin", geração de tíquetes (*Kerberos*, por exemplo) ou qualquer mudança de nível de permissão. É de especial importância observar acessos *anonymous* ou *guest* a servidores públicos.

5.16.2 Armazenando os dados auditados

Há três meios básicos de armazenagem de dados de auditoria. O primeiro é em arquivos comuns em meios magnéticos, tais como unidade de fitas, *pen drives*, discos rígidos em servidores, etc. O segundo é em mídia apenas de leitura como os DVD_R, por exemplo, e o terceiro é em papel impresso.

Obviamente, o meio mais inseguro é em mídia magnética, já que o invasor pode apagar seu rastro durante uma invasão. Por outro lado, é o meio mais barato e permite análise de dados por programas de auditoria. Dessa última vantagem, também goza a mídia apenas de leitura. O problema está no custo da mídia e do espaço necessário para sua armazenagem. Assim também é o caso do papel impresso.

Para todos os meios é imperativo manter segura a conexão entre o dispositivo que propriamente realiza o registro dos dados e o que os armazena.

5.16.3 Preservando dados de auditoria

A coleta de dados de auditoria pode resultar em rápido acúmulo de informação. A maneira mais eficiente parece ser a compressão de dados a partir de certo ponto no passado para trás, deixando apenas os dados recentes disponíveis para rápida consulta. É importante manter todos os detalhes nos arquivos comprimidos, já que pode ser necessário executar investigações minuciosas quando se descobre que algum tipo de incidente vinha ocorrendo por longo prazo.

É de vital importância manter *backups* das informações de auditoria, as quais devem ser guardadas como o máximo de segurança.

5.16.4 Questões legais

Para que os dados coletados sejam de utilidade em caso de demandas judiciais, é aconselhável procurar orientação legal para saber quais coletar. Se eles não forem definidos adequadamente antes de um incidente, isso pode resultar em falta de recursos para defesa.

Toda empresa que mantém dados de auditoria deve estar consciente de que

muitos desses dados podem conter informações pessoais cuja visualização, mesmo em caso de uma pesquisa de rotina pelo sistema de segurança, pode representar um caso de invasão de privacidade.

5.17 Lidando com incidentes de segurança

Num mundo de negócios competitivo como o de hoje, as empresas simplesmente não podem mais ficar indisponíveis para seus clientes, mesmo que tenham problemas com seus processos de negócios, recursos e/ou dados e informações. Velocidade de processamento e de decisões, altíssima disponibilidade, flexibilidade e foco em produtos de acordo com o mercado são requisitos fundamentais para "sobrevivência e sucesso". Porém, se não houver planejamento para segurança e contingência adequados, alguns ou até todos os requisitos estarão ameaçados e, consequentemente, a empresa também (LAUREANO, 2005).

Dessa maneira, é importante a rápida descoberta do incidente para o, mais rápido ainda, estabelecimento do sistema. Mas a primeira coisa a ser feita durante um incidente de segurança é manter a calma, pois ações desesperadas podem causar mais problemas do que o que já está feito.

É importante que os administradores de segurança percebam não só a necessidade de uma análise de rede, como também de procedimentos operacionais durante o incidente. A detecção da invasão traz novas necessidades de segurança, e junto com isso o conhecimento do que ocorre com os dados. Implantar um esquema de detecção de invasão requer planejamento.

5.17.1 Como detectar incidentes

Os indicativos de que há incidentes em curso podem ser os seguintes:

1. Sistema trava;
2. Novas contas de usuário ou atividade alta em contas anteriormente de baixa atividade;
3. Novos arquivos, muitas vezes com nomes estranhos;
4. Discrepâncias nos registros de conexão de usuários;
5. Mudanças no tamanho ou nas datas dos arquivos;

6. Tentativas de alterar arquivos;

7. Dados modificados ou apagados;

8. Negação de serviços;

9. Anomalias de funcionamento;

10. Sondagens externas suspeitas;

11. Navegações suspeitas em diretórios;

12. Incapacidade de um usuário entrar no sistema devido a mudanças nos dados de sua conta.

Essa lista não é única. Cada empresa pode constatar outros indicativos de incidentes de segurança. É importante determinar a extensão dos danos provocados pelo incidente. Por isso, as seguintes questões devem ser levantadas:

1. Este incidente ocorreu em outros *sites* também?

2. Quantos computadores em sua empresa foram afetados?

3. Alguma informação confidencial foi envolvida?

4. Por onde entrou o incidente (rede, linha telefônica, terminal local, etc.)?

5. A imprensa está envolvida?

6. Qual é o potencial de danos do incidente?

7. Qual é o tempo estimado para fechar o incidente?

8. Que recursos podem ser necessários para lidar com o incidente?

9. Alguma agência do governo está envolvida?

5.17.2 Trocas de informações

Quando há cooperação entre empresas ou instituições para o esclarecimento de uma invasão, é necessário trocar, no mínimo as seguintes informações:

1. Fuso horário dos registros (*logs*);

2. Informações sobre o sistema remoto, incluindo nomes de *hosts*, endereços IPs e (possivelmente) identificação de usuários;

3. todos os registros relevantes para o sistema remoto;

4. Tipo de incidente (o que aconteceu).

5.17.3 Protegendo as evidências

Ao responder a um incidente, documente todos os detalhes que tenham relação com ele, inclusive aqueles que vão lhe proporcionar economia de tempo durante a solução e erradicação do problema.

5.17.4 Contenção

O objetivo da contenção é limitar a extensão do ataque. Uma parte essencial para conter ataques envolve tomada de decisões tais como desligar ou não um sistema, desconectar ou não a rede, monitorar o sistema ou a atividade da rede, armar armadilhas, desabilitar funções tais como transferências remotas de arquivos, etc. A tomada dessas decisões deve ser estudada com antecedência e cuidadosamente para que as consequências não sejam piores que o ataque em si.

A contenção não deve ser confundida com a erradicação.

5.17.5 Erradicação

Antes de erradicar as causas do incidente, deve-se tomar cuidado em registrar todas as informações necessárias sobre o sistema comprometido, já que essas informações serão perdidas ao se restaurar o sistema.

Nesse processo podem ser usados programas tais como antivírus. É recomendável o armazenamento de programas suspeitos para análise, a reformatação dos discos infectados e análise do *backup*, para saber se não está infectado, antes de recuperá-lo.

Recomenda-se o uso dos registros para ajudar a erradicar a vulnerabilidade do sistema. Também é útil consultar com frequência os fabricantes dos programas e dos produtos de *hardware* para saber se há vulnerabilidades documentadas e se estas podem ser sanadas. Há *sites* na internet especializados em vulnerabilidades.

5.17.6 Recuperação

Uma vez erradicadas as causas, a fase de recuperação define o próximo passo. O objetivo é retornar o sistema à operação normal, de preferência, com o mínimo de incômodo para o usuário.

5.17.7 Plano de contingência

O planejamento da segurança pode ser resumido conforme as declarações a seguir:

1. Descobrir como aconteceu o incidente de segurança;
2. Descobrir como evitar a exploração da mesma vulnerabilidade;
3. Evitar a escalada de mais incidentes;
4. Avaliar o impacto e os danos do incidente;
5. Recuperar-se do incidente;
6. Atualizar as políticas e procedimentos conforme a necessidade;
7. Descobrir quem fez (se apropriado e possível).

O passo número 7 acima é confirmar a identidade do invasor e depois verificar com provas se ele realmente invadiu a rede. Pode-se estabelecer uma série de perguntas ao provável invasor com o objetivo claro e explícito, especialmente para o suspeito, de confirmar a invasão. Caso seja confirmada pelo usuário, pode-se solicitar explicação sobre o procedimento, proporcionando condição para e aplicar as sanções da política de segurança da empresa. Em alguns casos pode ser necessário comunicar o fato às autoridades competentes. Em caso de negação do usuário, uma investigação maior deverá ser levada a cabo, antes que se possa fazer qualquer tipo de acusação.

Dependendo do plano de ação estabelecido para os casos de invasão, pode-se negar acesso ao servidor do invasor ou a todos os usuários que venham de fora da rede, ou travar todas as contas de usuário ou chegar ao ponto de matar todos os processos de usuários e reinicializar os servidores.

Ao determinar o plano de contingência em caso de incidentes de segurança,

os administradores podem se ver na frente de vários dilemas. Se um sistema é crítico e deve permanecer ativo, ao restaurá-lo, por exemplo, recuperando um *backup*; pode-se apagar os eventos que poderiam ajudar na análise posterior do incidente. Portanto, é necessário não só considerar tais ocorrências como determinar as várias prioridades dentro do plano de contingência.

5.17.8 Prioridades das defesas

São prioridades da defesa:

1. Proteger a vida humana e a integridade física das pessoas;

2. Proteger dados sigilosos;

3. Proteger dados tais como proprietários, científicos, gerenciais;

4. Impedir danos aos sistemas;

5. Minimizar a indisponibilidade de recursos computacionais.

Qualquer plano para responder a incidentes de segurança deve ser guiado pelas normas e legislação local.

5.17.9 Acompanhamento posterior

Depois de recuperar sistema, é possível que ainda haja brechas e até mesmo armadilhas à espreita no sistema. Por isso deve haver um acompanhamento posterior a um incidente. Esse acompanhamento envolve a monitoração do sistema para itens não observados durante a fase de erradicação.

Como parte dos procedimentos de acompanhamento posterior, é importante produzir um histórico do incidente, desde sua detecção até a erradicação, de modo a servir de base para solução de problemas futuros.

5.17.10 Cálculo dos prejuízos

Após sair de um incidente, devem ser tomadas as seguintes providências:

1. Um levantamento cuidadoso dos ativos do sistema deve ser realizado (ou seja, um exame cuidadoso para saber como o sistema foi afetado pelo incidente);

2. As lições aprendidas como resultados dos incidentes devem ser incluídos na revisão do plano de segurança, para evitar que o incidente ocorra novamente;

3. Sob a luz do incidente, deve-se desenvolver uma nova análise de risco.

5.17.11 Responsabilidades

Ao proteger o sistema contra invasão externa, o administrador de segurança pode ser tentado a rastrear a invasão até sua origem, passando por redes alheias. Essa prática pode caracterizar uma invasão nessas redes. Nem sempre o invasor é de fato parte dela. Ele pode estar usando a rede alheia como trampolim para executar um ataque.

De outro modo, é preciso saber se a invasão em curso na rede está sendo usada para invadir outras redes. Se isso for verificado, é importante avisar os administradores das redes envolvidas de forma a demonstrar uma política de cooperação mútua na busca da origem da invasão.

5.18 Tarefas constantes

Veja a cartilha do NicBr para navegar seguro pela internet em http://cartilha.cert.br/

Leia a Reportagem da revista veja sobre ataques do grupo de hackers LulzSec em http://revistaepoca.globo.com/Revista/Epoca/0,,EMI243905-15227,00-A+GUERRA+VIRTUAL+COMECOU.

Conheça "O Futuro da Internet" acessando http://olhardigital.uol.com.br/jovem/digital_news/consegue-prever-o-futuro-da-internet

Veja "Casa automatizada: mais segurança ou mais riscos de invasão" em http://olhardigital.uol.com.br/colunistas/jose_milagre/post/casa_automatizada_mais_seguranca_ou_mais_riscos_de_invasao

É muito importante ter a noção de que a preocupação com a segurança não acaba nunca. Ela tem de ser uma atividade contínua, pois pior que ter um sistema inseguro é ter um sistema falsamente seguro passando uma falsa impressão de segurança.

Por isso, estar atento aos seguintes itens será de grande valia nessa tarefa de manter os sistemas:

1. Assine boletins periódicos editados pelas equipes de resposta a incidentes de segurança, tais como o *CERT Coordination Center*, e atualize seus sistemas contra as ameaças que se aplicam à tecnologia de sua rede;

2. Acompanhe o lançamento de *patches* (remendos) de segurança produzidos pelos fabricantes de seus equipamentos, adquira-os e instale os que se aplicarem;

3. Observe ativamente as configurações de seus sistemas, identifique quaisquer mudanças ocorridas e investigue todas as anomalias;

4. Reveja todas as políticas e procedimentos de segurança pelo menos uma vez por ano;

5. Assine listas de discussão relevantes para se manter atualizado com as informações mais recentes compartilhadas por colegas administradores;

6. Verifique regularmente a obediência às políticas e procedimentos de segurança. Esta auditoria deve ser realizada por pessoas diferentes daquelas que definem e implementam as políticas e procedimentos;

7. Evite senhas fáceis de serem descobertas;

8. Usar um bom antivírus sempre atualizado;

9. Use um bom *firewall*;

10. Use um bom *antispyware,* que é um programa que evita que programas espiões se instalem no seu computador e roubem suas informações. Ele também pode localizar *spywares* já instalados e destruí-los;

11. Coloque sempre o navegador com as opções de segurança em "nível alto". Um navegador seguro é aquele que lhe permite navegar pela internet sem que *sites* maliciosos consigam coletar seus dados, instalar *spywares* ou outros programas indesejáveis e potencialmente danosos no seu computador.

12. Tome cuidado com a engenharia social, evitando sempre fornecer dados pela internet como telefone, endereço, senhas (como a de acesso ao seu provedor), número do seu cartão de crédito, CPF, *e-mail*, etc.;

13. Realize cópias de segurança (*backups*) dos dados armazenados em um computador, as quais são importantes não só para se recuperar de eventuais falhas, mas também das consequências de uma possível infecção por vírus ou de uma invasão;

14. Use sempre a criptografia dos dados. Os dados armazenados em uma cópia de segurança podem conter informações sigilosas. Por isso eles devem ser armazenados em algum formato criptografado.

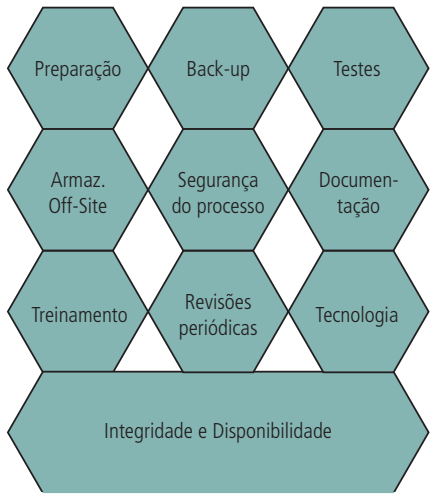


Figura 5.15: Dicas de segurança
Fonte: www.mlaureano.org/aulas_material/gst/apostila_versao_20.pdf

Resumo

Nesta aula você viu os processos e sistemas de segurança dentro do contexto de uma visão ampla de políticas de segurança. Conheceu os risco e vulnerabilidades dos sistemas computacionais bem como as técnicas e mecanismos de defesa.

Atividades de aprendizagem

1. Qual o nome que se dá à técnica de ataque de Enchimento de Tráfego, que é caracterizada pelo envio maciço de informações a um servidor?

2. Explique o que é o Registro de Eventos.

3. O que significa criptografar a mensagem "TRANSMITIR CODIGO"? Dê um exemplo dessa mensagem criptografada.

4. O que é criptografia assimétrica?

5. O que é risco?

6. O que é uma vulnerabilidade?

7. A ação de um *hacker* provavelmente consiste em uma sequência de ataques, que pode ser dividida em três fases:

    - Reconhecimento (coleta de informação)

- Acesso (caminho de invasão)
- Ação efetivamente danosa

8. Enquadre nessas três fases um ou mais dos seguintes exemplos:

- *Back door*
- *Port scanning*
- *Spoofing* (ex.: IP *spoof*)
- Estouro de pilha
- Quebra de senha
- *Sniffing*
- Desfiguração de *website*
- *War dialing*

Poste suas respostas no AVEA!

Referências

ALBUQUERQUE, F. **TCP/IP – Internet: protocolos e tecnologias**. Rio de Janeiro: Axcel Books, 2001.

BARAN, M.; WU, F. F. Optimal sizing of capacitors placed on a radial distribution system. **IEEE Transactions on Power Delivery**, v. 4, n. 1, p. 735-743, jan. 1989.

BATTISTI, Julio. **Redes e TCP/IP**. Disponível em: <http://www.juliobattisti.com.br/artigos/redes.asp/>. Acesso em: 2 nov. 2012.

BERNAL, Volnys Borges. **Tecnologia de redes:** protocolo Ethernet. Disponível em: <www.lsi.usp.br/~volnys/courses/tecredes/pdf/05ETH-col.pdf>. Acesso em: 1 nov. 2012.

BRASIL. Ministério do Planejamento, Orçamento e Gestão. **Gestão de domínios**. Disponível em: <http://www.governoeletronico.gov.br/acoes-e-projetos/gestao-de-dominios>. Acesso em: 1 nov. 2012.

CARVALHO, Márcio Luiz Bunte. **História da internet no Brasil**. Disponível em: <http://homepages.dcc.ufmg.br/~mlbc/cursos/internet/historia/Brasil.html>. Acesso em: 1 nov. 2012.

CASTRO, Maria Cristina F. et al. **Redes comutadas**. 2002. Disponível em: <http://www.feng.pucrs.br/~decastro/download.html>. Acesso em: 4 nov. 2012.

CERT.BR. Núcleo de informação e Coordenação do ponto BR. <**Cartilha de segurança na internet**. Disponível em: <http://cartilha.cert.br/livro/cartilha-seguranca-internet.pdf>. Acesso em: 4 nov. 2012.

CINTRA, Glauber Ferreira. **Perl é uma linguagem de programação baseada, principalmente, nas linguagens C e AWK**. 2001. Disponível em: http://www.ime.usp.br/~glauber/perl/perl.htm. Acesso em: 1 nov. 2012.

CISCONET. Sharing. Cisco Expertise. **TCP/IP**. Disponível em: <http://cisconet.com/tcpip.html>. Acesso em: 2 nov. 2012.

COMER, Douglas E. **Interligação em redes com TCP/IP**. 5. ed. Porto Alegre: Bookman, 2006.

COMER, Douglas E. **Redes de computadores e internet**. 4. ed. Porto Alegre: Bookman, 2007.

COUTINHO, Bruno Cardoso. **Redes TCP/IP: curso técnico em informática** Colatina: CEAD/ Ifes, 2010; Vitória: Etec Brasil, 2010.

CURVELLO, Rodrigo. **Sistemas de conectividade.** Olinda:Veneza Livros, 2007.

CYCLADES. **Guia internet de conectividade**. 6. ed. Brasília: SENAC, 2000.

FILIPPETTI. Marco. **Tutorial VLSM (Variable Lenght Subnet Mask)**. 2008. Disponível em: <http://blog.ccna.com.br/2008/01/05/tutorial-vlsm-variable-lenght-subnet-masks/>. Acesso em: 2 nov. 2012.

GAI, Silvano. **Internetworking IPv6 With Cisco Routers**. São Paulo: Mcgraw-Hill, 1998.

GONÇALVES, José. **Equipamentos de interconexão:** hubs, pontes e switches. Disponível em: <www.inf.ufes.br/~zegonc/material/S.O.%20II/Switching.pdf>. Acesso em: 4 nov. 2012.

KUROSE, James F. et al. **Redes de computadores e a internet.** 5. Ed. São Paulo: Pearson, 2010.

KUWABARA, Marcelo. **Instrutor eletrônico**. Disponível em: <http://www.oocities.org/siliconvalley/hub/5612/>. Acesso em: 1 nov. 2012.

LAGES, Walter Fetter. **Camadas de transporte**. Disponível em: <http://www.ece.ufrgs.br/~fetter/ele00012/transporte.pdf>. Acesso em: 1 nov. 2012.

LAUREANO, Marcos Aurélio P. **Gestão de segurança da informação**. 2005. Disponível em: http://www.mlaureano.org/aulas_material/gst/apostila_versao_20.pdf. Acesso em: 4 nov. 2012.

LIMA, Isaias. **Apostila de rede de computadores**. Universidade Federal de Itajubá. Disponível em: <http://pt.scribd.com/doc/51125905/146/Endereco-de-Loopback>. Acesso em: 2 nov. 2012.

MAIA, Luís P. **Arquitetura de redes de computadores.** Rio de Janeiro: LTC, 2009.

MELO, Sandro. **Exploração de vulnerabilidade em redes TCP/IP**. 2. ed. São Paulo: Editora Alta Books, 2006.

MORALES, Jonathan. **VLAN**. Disponível em: <http://www.ciscoredes.com/ccna3/90-vlan.html>. Acesso em: 4 nov. 2012.

NAKAMURA, Emílio T. et al. **Segurança de redes em ambientes cooperativos**. São Paulo: Novatec Editora, 2007.

PEREIRA, José Gonçalves. **VLSM e CIDR**. Disponível em: http://www.inf.ufes.br/~zegonc/material/S.O.%20II/VLSM%20e%20CIDR%20(1pag).pdf. Acesso: 02 nov. 2012.

PINHEIRO, José Maurício dos Santos. **Switches em redes locais de computadores**. 1995. Disponível em: <http://www.projetoderedes.com.br/artigos/artigo_switches_em_redes_locais.php>. Acesso em: 4 nov. 2012.

REGISTROBR. **Info: Dicas e regras**. Disponível em: <http://beta.registro.br/info/dicas.html>. Acesso em: 1 nov. 2012.

RIOS, Harley F. R. **Projeto de redes de computadores.** 2011. Disponível em: <http://pt.scribd.com/doc/95940234/56/Roteadores>. Acesso em: 4 nov. 2012.

SANTOS, Rodrigo Regis et al. **Curso IPv básico**. Núcleo de Informação e Coordenação do Ponto BR, São Paulo. 2010. Disponível em: <http://ipv6.br/download/IPv6-apostila.pdf>. Acesso em: 3 nov. 2012.

SENGER, Herme. **Modelo de referência OSI**. Disponível em: <http://www-usr.inf.ufsm.br/~candia/aulas/espec/Aula_3_Modelo_OSI.pdf>. Acesso em: 31 out. 2012.

SIMON, Imre. **Hipertexto**. Disponível em: <http://www.ime.usp.br/~is/abc/abc/node9.html>. Acesso em: 1 nov.2012.

COUTINHO, Bruno Cardoso. **Sistemas operacionais**: Colatina: CEAD /Ifes, 2010. Curso Técnico em Informática.

SOARES, Luiz F.G. **Redes de computadores:** das Lan's, Man's e Wan's às redes ATM. 2ª ed. Rio de Janeiro: Campus, 1995.

TANENBAUM, Andrew. S. **Redes de computadores**. 4ª ed. Rio de Janeiro: Campus, 2003.

TAPAJÓS, Mauro. **Introdução às redes de comunicação de dados**. Disponível em: <http://pt.scribd.com/doc/77020951/18/Tipos-de-Comutacao-switching>. Acesso em: 4 nov. 2012.

TORRES, Gabriel. **Redes de computadores**. Versão revisada e atualizada. Rio de Janeiro: Nova Terra, 2009.

TYSON, Jeff. **Como funcionam os switches LAN (rede de comunicação local)**. Disponível em: <http://informatica.hsw.uol.com.br/lan-switch16.htm>. Acesso em: 4 nov. 2012.

ULBRICH, Henrique César; VALLE, James Della. **Universidade hacker.** 2ª ed. São Paulo: Digerati, 2003.

VALLE, Maria do Socorro Costa et al. **Internet:histórico, evolução e gestão.** Disponível em: http://www.rederio.br/downloads/pdf/nt00501.pdf>. Acesso em: 1 nov. 2012.

Currículo do professor-autor

Marcos Prado Amaral

Possui graduação em Engenharia Elétrica pela Pontifícia Universidade Católica de Minas Gerais (PUC/MG) e mestrado em Tecnologia pelo Centro Federal de Educação Tecnológica de Minas Gerais (CEFET/MG). Atualmente é professor do 1º e 2º graus do CEFET/MG na modalidade presencial e a distância. Tem experiência na área de Ciência da Computação, com ênfase em Sistemas de Computação, atuando principalmente nos seguintes temas: Rede de Computadores, Ciência e Tecnologia, Sistemas Operacionais, Segurança da Informação e Informática.

rede e-Tec Brasil

ISBN 978-85-99872-23-9