RESUMO TCP/IP – PROFESSOR ANTONIO EUGENIO

1. HISTÓRICO

A arquitetura TCP/IP surgiu com a criação de uma rede patrocinada pelo Departamento de Defesa do governo dos Estados Unidos da América (DoD - *Department of Defense*). Uma das tarefas essenciais dessa rede seria manter comunicados, mesmo que apenas uma parte, órgãos do governo e universidades, numa ocorrência de guerras ou catástrofes que afetassem os meios de comunicação daquele país. Dessa necessidade, surgiu a ARPANET, uma rede que permaneceria intacta caso um dos servidores perdesse a conexão.

A ARPANET necessitava então de um modelo de protocolos que assegurasse tal funcionalidade esperada, mostrando-se confiável, flexível e de fácil implementação. É então desenvolvida a arquitetura TCP/IP, que se torna um padrão de fato.

A ARPANET cresceu e tornou-se a rede mundial de computadores - Internet. A utilização (e facilidades) do padrão TCP/IP utilizado pelos fabricantes de outras redes, com a finalidade da conectividade com a Internet. A normalização do TCP/IP chegou após a sua utilização em massa.

Assim, o TCP/IP (*Transmission Control Protocol/Internet Protocol*, ou Protocolo de Controle de Transmissão / Protocolo da Internet) se refere ao conjunto de protocolos utilizados na Internet. Ele inclui uma série de padrões que especificam como os computadores vão se comunicar e cria convenções para interconexão de redes e para o roteamento através dessas conexões.

Ele foi utilizado em todas as redes de longa distância do sistema de Defesa dos EUA em 1983, mas não foi amplamente aceito até ser incorporado ao BSD (*Berkeley Software Distribution*) Unix 4.2. A popularidade do TCP/IP é baseada em:

- Estrutura cliente-servidor robusta. O TCP/IP é uma excelente plataforma cliente-servidor, especialmente em ambientes WAN (*wide area Network*, ou redes de grande alcance).

- Compartilhamento de informações. Milhares de organizações militares, educacionais, científicas e comerciais compartilham dados, correio eletrônico (*e-mail*), e outros serviços na Internet usando o TCP/IP.

- Ampla disponibilidade. Implementações do TCP/IP estão disponíveis em praticamente todos os sistemas operacionais populares. Seu código fonte é amplamente disponível em várias implementações. Fabricantes de *bridges*, *routers* e analisadores de redes oferecem suporte para o TCP/IP em seus produtos.

2 . CONCEITOS BÁSICOS

Gateway Padrão (*Default Gateway*)

O *gateway* padrão é a máquina para quem os hosts solicita ajuda quando não consegui achar uma outra máquina na rede. Funciona assim: Quando uma máquina na rede precisa se comunicar com uma outra, ela emite um pedido de conexão (esse pedido é feito através de *broadcasting*, ou seja, a máquina envia um pedido a toda a rede, e apenas a máquina destino responde) e aguarda uma resposta. Se a resposta não vier, ela entra em contato com o *gateway* padrão e solicita que o mesmo conecte com a máquina destino. Se o *gateway* conseguir se conectar à máquina destino, ele fica como "intermediador" dessa conexão, caso contrário ele avisa a máquina solicitante que não foi possível encontrar a máquina destino.

Domain Name System (DNS)

Domain Name System (ou Sistema de Nomes de Domínio). Essas 3 palavras também não significam muita coisa para a maior parte das pessoas também, por isso vamos à analogia com o telefone. Quando deseja telefonar para a loja da esquina, você consulta o catálogo, descobre o telefone de lá e liga. Você não consegue telefonar para lugar algum se não souber o número do telefone. Na rede TCP/IP acontece a mesma coisa. Os usuários não decoram o número IP das máquinas, e sim seus nomes. Mas para se alcançar uma máquina na rede, precisamos do seu número de IP. Para resolver isso, foi criado o DNS, um serviço disponível na rede que, dado um nome de máquina, ele retorna o número de IP da mesma.

O Windows NT oferece um serviço semelhante, o WINS (*Windows Internet Name System*, ou Sistema de nomes da Internet do Windows). A principal diferença entre os dois é que o DNS usa uma tabela estática, e o WINS usa uma tabela dinâmica. No caso do servidor DNS rodar numa máquina Windows NT é recomendável que ele seja substituído pelo WINS.

Dynamic Host Configuration Protocol (DHCP)

DHCP significa *Dynamic Host Configuration Protocol* (Protocolo de Configuração de *Host* Dinâmico). Numa rede TCP/IP, todo computador tem de ter um número de IP distinto. Isto significa que antes de colocar uma nova máquina na rede, o administrador teria de checar quais números estão sendo utilizados para poder escolher um n.º adequado para a nova máquina. Em pequenas redes isso é possível de ser feito, mas em grandes redes isso se torna uma tarefa muito tediosa e sujeita a falhas. Para evitar isso, foi criado o DHCP. Quando uma máquina entra na rede, ela procura o servidor DHCP (cujo n.º de IP foi previamente fornecido) e solicita um n.º de IP para si própria. O servidor verifica qual o n.º disponível, informa ao solicitante esse n.º e o torna indisponível para futuras solicitações. Dessa maneira, a administração dos n.º de IP é feita automaticamente e não existem problemas de conflito. Quando a máquina solicitante sai da rede, o servidor DHCP torna seu n.º de IP disponível novamente.

Portas

Uma porta pode ser vista como um canal de comunicações para uma máquina. Pacotes de informações chegando a uma máquina não são apenas endereçadas à maquina, e sim à máquina numa determinada porta. Você pode imaginar uma porta como sendo um canal de rádio, com a diferença fundamental de que um computador pode "ouvir" a todos os 65000 canais possíveis ao mesmo tempo!
Entretanto, um computador geralmente não está escutando a todas as portas, ele escuta umas poucas portas específicas. E ele não vai responder a um pedido que chegue numa porta a qual ele não esteja escutando.

Existem uma série de portas pré-definidas para certos serviços que são aceitos universalmente. As principais são:

Serviço	Porta	Descrição
FTP	**21**	***File Transfer Protocol* (Protocolo de Transferência da Arquivos)**
TELNET	**23**	**Para se conectar remotamente a um servidor**
SMTP	**25**	**Para enviar um *e-mail***
Gopher	**70**	***Browser* baseado em modo texto**
HTTP	**80**	**Protocolo WWW – *Netscape, Mosaic***
POP3	**110**	**Para receber *e-mail***
NNTP	**119**	***Newsgroups***
IRC	**6667**	***Internet Relay Chat* – Bate papo on-line**
Compuserve	**4144**	***Compuserve WinCIM***
AOL	**5190**	***America Online***
MSN	**569**	**Microsoft Network**

SOCKETS

Sockets, introduzido pelo UNIX de *Berkeley*, é um mecanismo básico de identificação de uma aplicação específica em todo domínio da Internet ou entre comunicações interprocessos (domínio UNIX). O *socket* é o ponto final de uma comunicação na rede. O TCP multiplexa multiplas conexões para um simples *host* usando *sockets* e portas. O *socket* é a concatenação do endereço de rede (IP) com o número da porta. Um *socket* não é uma porta, embora exista uma relação entre eles (um para muitos). Cada porta pode ter um simples *socket* passivo, esperando por uma conexão entrante, e múltiplos *sockets* ativos, cada um correspondendo a uma conexão aberta na porta. Assim, um *socket* local pode participar de varias conexões com vários *sockets* remotos. O encerramento de uma conexão desativa os *sockets* em cada extremidade.

Assim, para que os dados enviados por uma aplicação possam ser entregues para a aplicação correta no computador destino, é necessário algum método de identificar as próprias aplicações origem e destino, além de simplesmente identificar os computadores origem e destino através dos endereços IP - Para que as aplicações possam ser identificadas, cada aplicação tradicional do TCP/IP (FTP, E-MAIL, etc) tem um endereço próprio de porta ("*Well-Known Socket*") previamente conhecido que a identifica.
Ou seja, além dos endereços IP que identificam os *hosts*, é necessária uma forma qualquer de identificar as próprias aplicações clientes e servidoras envolvidas na comunicação.

Quando uma mensagem é passada por uma aplicação qualquer para o TCP transportar, além do endereço IP de destino, os códigos que identificam a aplicação origem (que está enviando a mensagem) e a aplicação destino (a aplicação para a qual a mensagem deve ser entregue no computador destino) também são armazenados no cabeçalho do datagrama gerado pelo TCP.

Note que as aplicações clientes TCP/IP (que requisitam serviços) podem usar endereços ou números de portas aleatórias, mas as aplicações servidoras (as que fornecem serviços baseados no TCP/IP para as aplicações clientes), como servidores de FTP e de *E-Mail*), rodam em portas bem definidas, cujos números são padronizados em *RFCs*, para que possam ser acessadas com facilidade por qualquer usuário na Internet.

A aplicação cliente não precisa utilizar um número de porta conhecido porque ninguém tentará conectar-se a ela para obter serviços! As aplicações servidoras, entretanto, devem utilizar portas conhecidas (porta 21 no caso do FTP), para que as pessoas possam abrir conexões e começar a enviar comandos e dados para elas.

Por exemplo, se você quer obter um arquivo pela Internet utilizando o protocolo FTP, você deve carregar em sua estação uma aplicação "cliente" de FTP. Esta aplicação abre uma conexão TCP utilizando como endereço de porta origem um número aleatório, por exemplo, 1278. Entretanto, a aplicação cliente deve especificar como endereço de porta destino o número 21, que é o número oficial da porta para as aplicações servidoras de FTP.

Com base no que foi explicado nos parágrafos anteriores, podemos compreender melhor o conceito de que uma conexão TCP é na verdade identificada por um grupo de 4 números:

•o endereço Internet do computador origem (por exemplo, 175.11.1.4);
•o endereço Internet do computador destino (por exemplo, 200.230.55.10);
•o número da porta origem, utilizado pela aplicação cliente (exemplo, 1278);
•o número da porta destino da aplicação no servidor (exemplo, 21 para o FTP).

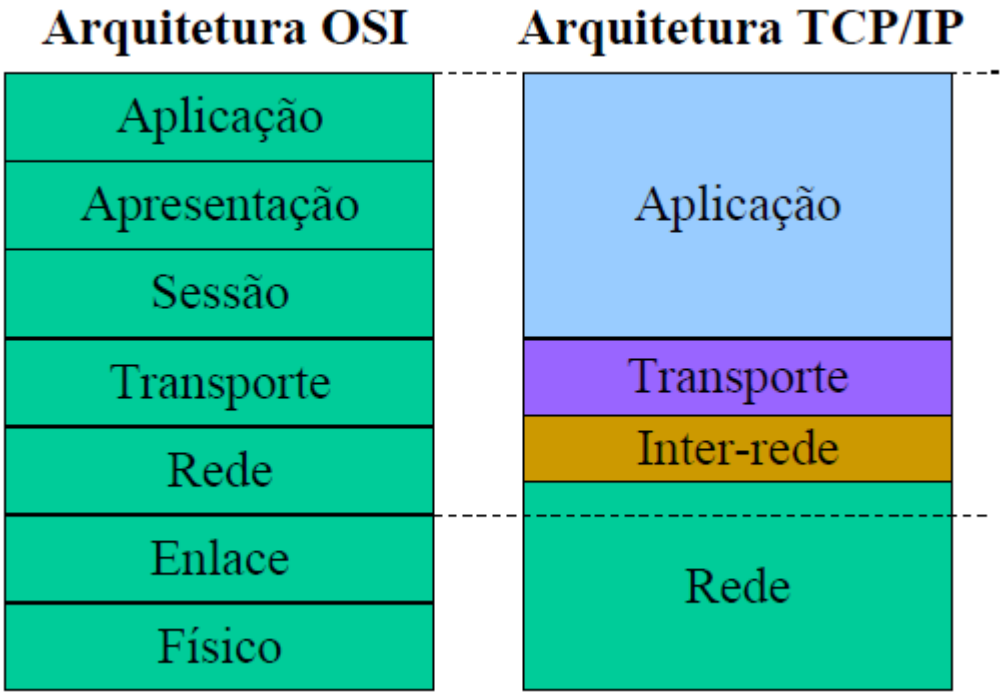
3. MODELO TCP/IP

O modelo TCP/IP é constituído basicamente por duas (02) camadas: a camada de rede e a camada de transporte. Tanto a camada de aplicação quanto a camada de interface de rede não possuem uma norma definida, devendo a camada de aplicação utilizar serviços da camada de transporte, a ser definida adiante, e a camada de interface de rede prover a interface dos diversos tipos de rede com o protocolo (promovendo em conseqüência a interoperação entre as diversas arquiteturas de rede — Ethernet, Token Ring, ATM, etc.

A camada de rede é a primeira (normatizada) do modelo. Também conhecida como camada Internet, é responsável pelo endereçamento, roteamento dos pacotes, controle de envio e recepção (erros, bufferização, fragmentação, seqüência, reconhecimento, etc.). Portanto, o IP é um protocolo que providência a entrega de pacotes para todos os outros protocolos da família TCP/IP.

A arquitetura TCP/IP possui uma série de diferenças em relação à arquitetura OSI. Elas se resumem principalmente nos níveis de aplicação e Inter-rede da arquitetura TCP/IP.
Como principais diferenças pode-se citar:
• OSI trata todos os níveis, enquanto TCP/IP só trata a partir do nível de Rede OSI
• OSI tem opções de modelos incompatíveis. TCP/IP é sempre compatível entre as várias implementações
• OSI oferece serviços orientados a conexão no nível de rede, o que necessita de inteligência adicional em cada equipamento componente da estrutura de rede. Em TCP/IP a função de roteamento é bem simples e não necessita de manutenção de informações complexas.
• TCP/IP tem função mínima (roteamento IP) nos nós intermediários (roteadores)
• Aplicações TCP/IP tratam os níveis superiores de forma monolítica, desta forma OSI é mais eficiente pois permite reaproveitar funções comuns a diversos tipos de aplicações. Em TCP/IP, cada aplicação tem que implementar suas necessidades de forma completa.

A figura abaixo ilustra a comparação entre TCP/IP e OSI. Note que a camada Inter-rede de TCP/IP apresenta uma altura menor que o correspondente nível de Rede OSI. Isto representa o fato de que uma das funções do nível de Rede OSI é realizada pelo nível de Rede TCP/IP. Esta função é a entrega local de mensagens dentro da mesma rede. O IP só trata a entrega e a decisão de roteamento quando a origem e o destino da mensagem estão situados em redes distintas.



O IP oferece um sistema de entrega de dados sem conexão, sendo a comunicação através de datagramas. Isto é, os pacotes IP não são garantidos de chegarem ao seu destino, nem de serem recebidos na ordem em que foram enviados. O checksum do IP confirma apenas a integridade do cabeçalho do pacote. Desta maneira, a responsabilidade pelos dados contidos no pacote do IP (e sua seqüência) é tarefa de protocolos de mais alto nível.

- **Formato dos datagramas IP**

0		7	15	23	31
Versão	IHL	Tipo de Serviço	Comprimento Total		
IDENTIFICAÇÃO			*FLAGS*	*OFFSET* de Fragm.	
Tempo de Vida		Protocolo	*Checksum* do Cabeçalho		
Endereço de Origem					
Endereço de Destino					
Opções IP					*Padding*
Dados					
Dados					

- **Versão**
  Indica a versão do protocolo IP

- **IHL**
  Comprimento do cabeçalho IP, para indicar o comprimento em número de palavras de 32 *bits*, do campo versão ao inicio do campo de dados. Sendo que o comprimento mínimo é de 5 palavras.

- **Tipo de serviço**
  Fornece uma indicação dos parâmetros da qualidade de serviço desejada. Estes parâmetros são usados como orientação na seleção dos serviços de transmissão de dados das sub-rede por onde os datagramas passarão.

- **Comprimento Total**
  Fornece o comprimento total do datagrama, em octetos, do campo "VERSÃO" a parte de dados inclusive. O comprimento máximo é de 64 Kbytes. Na prática entretanto, utiliza-se 576 octetos.

- **Identificação**
  É usado na montagem dos fragmentos de um datagrama. O campo "*flags*" (3 bits) serve de controle de fragmentação, indicando se um datagrama pode ou não ser fragmentado e se houve fragmentação. O campo *offset* de fragmento (13 bits) indica o posicionamento do fragmento dentro do datagrama original. Este posicionamento é medido em unidades de 8 octetos. Este campo vale zero em datagramas não fragmentados e no primeiro fragmento de um datagrama.

  O campo *flags* do cabeçalho do datagrama IP é composto dos bits DF (*don´t fragment*) e MF (*more fragments*). Quando DF=1, os roteadores não podem fragmentar o datagrama.

- **Tempo de Vida**
  Indica o tempo máximo que o datagrama pode trafegar em uma rede *Internet*, sendo este campo decrementado em cada *gateway*. Quando seu conteúdo chega a zero, o datagrama é descartado.
  O objetivo é descartar datagramas que não atingem o seu destinatário.

- **Protocolo**
  Identifica o protocolo de transporte que gerou e que irá receber o datagrama.

- **Checksum**
  Serve para identificar erros ocorridos durante a transmissão ou na atualização do cabeçalho; desta forma, o *checksun* é recalculado e verificado a cada ponto onde o cabeçalho é processado (nos *gateways*).

- **Endereços de origem e destino**
  Carregam os endereços IP's da fonte e do destino.

- **Opções**
  É usado para informações de segurança, relatórios de erro, controle, testes ou pode estar vazio.

- **Padding**
  É usado para garantir que o comprimento do cabeçalho seja múltiplo inteiro de 32 *bits*.

Dentre os protocolos da Camada de Rede, destaca-se inicialmente o IP (*Internet Protocol*), além do ARP, ICMP, RARP e dos protocolos de roteamento (RIP ,IGP, OSPF, Hello, EGP e GGP). Demais informações a respeito dos protocolos desta camada, serão descritas adiante.

Talvez o protocolo de alto nível do IP mais comum seja o TCP. O TCP oferece um confiável protocolo baseado em conexão encapsulado no IP. O TCP garante a entrega dos pacotes, assegura o sequenciamento dos pacotes, e providencia um checksum que valida tanto o cabeçalho quanto os dados do pacote. No caso da rede perder ou corromper um pacote TCP/IP durante a transmissão, é tarefa do TCP retransmitir o pacote faltoso ou incorreto. Essa confiabilidade torna o TCP/IP o protocolo escolhido para transmissões baseadas em sessão, aplicativos cliente-servidor e serviços críticos como correio eletrônico.

Porém essa confiabilidade tem um preço. Os cabeçalhos dos pacotes TCP requerem o uso de bits adicionais para assegurar o correto sequenciamento da informação, bem como um *checksum* obrigatório para garantir a

integridade do cabeçalho e dos dados. Para garantir a entrega dos pacotes, o protocolo também requer que o destinatário informe o recebimento do pacote.

Tal "informação de recebimento" (ou ACKs, de *acknowledgments*) geram tráfego adicional na rede, diminuindo a taxa de transferência de dados em favor da confiabilidade. Para reduzir o impacto na performance, a maioria dos servidores enviam um ACK para todo segmento de dados (ao invés de todo pacote) ou quando um ACK expira.

ARP e ICMP

Dois outros protocolos na família TCP/IP tem importantes funções, embora essas funções não estejam diretamente relacionadas com a transmissão de dados: ARP (*Address Resolution Protocol*, ou Protocolo de Resolução de endereços) e ICMP (*Internet Control Message Protocol*, ou Protocolo de Controle de Mensagens da Internet). O ARP e o ICMP são protocolos de manutenção que mantêm a estrutura do IP e usualmente são invisíveis aos usuários e às aplicações.

O cabeçalho do IP contém tanto o endereço IP da origem quanto do destino, mas o endereço do hardware também tem de ser conhecido. O IP obtém um endereço de hardware de um determinado sistema difundindo pela rede um pacote especial de requisição (um pacote ARP de requisição) contendo o endereço IP do sistema com o qual está tentando se comunicar. Todos os nós da rede local que tiverem o ARP habilitado detectam essa difusão, e o sistema que tem o número de IP em questão envia um pacote (do tipo *ARP reply*, ou resposta ARP) contendo seu endereço de hardware para o computador que o solicitou. O endereço de hardware e o endereço IP do computador estão é armazenado no cache do ARP para uso futuro. Como a resposta ARP também é feita na forma de difusão, é normal que outros nós usem essa informação para atualizar seus caches ARP.

O ICMP permite que 2 nós em uma rede IP compartilhem o status do IP (protocolo) e informação de erros. Esta informação pode ser usada por protocolos de alto nível para tratar problemas de transmissão ou para administradores de rede para detectar problemas na rede. Embora estejam encapsulados em pacotes IP, o ICMP não é considerado um protocolo de alto nível (ele é necessário em toda implementação do TCP/IP). O utilitário ping faz uso do ICMP para determinar se um certo endereço IP na rede está operacional. Isto é útil para diagnosticar problemas em redes IP ou falhas em gateways.

As ocorrências do ICMP podem ser:

a. destinatário inacessível;
b. ajuste de fonte — Solicita à estação a redução da taxa de emissão de datagramas;
c. redireção — Rota mais adequada para a estação destinatária (para atualização da tabela de endereço dos roteadores;
d. eco e Resposta de Eco;
e. tempo excedido;
f. Problemas de parâmetros;
g. marca de Tempo e Resposta;
h. solicitação de informações e Respostas de Informações;
i. solicitação de Máscara de endereço e Resposta à Mascara de Endereço.

4. Outros Protocolos:

Além desses protocolos citados, existem os protocolos de alto nível, como o Telnet, FTP, HTTP e etc. Abaixo uma breve descrição deles :

• Telnet : É um protocolo que permite o logon em máquinas remotas. Você passa a utilizar a máquina remota para realizar o processamento. No Windows NT existe o RAS (*Remote Access Service*, Serviço de Acesso Remoto) que tem os mesmos objetivos do Telnet.

• FTP : *File Transfer Protocol* (protocolo de transferência de arquivos), como o nome já diz é utilizado para a transferência de arquivos.

• HTTP: *Hyper Text Transfer Protocol* : É o protocolo utilizado pela Web, ele transmite textos, gráficos e qualquer outro tipo de arquivo (substituindo o FTP) além de permitir a navegação através de hiper texto.

5. Camada de Transporte

A camada de transporte é uma camada fim-a-fim, isto é, uma entidade desta camada só se comunica com a sua entidade par do host destinatário. É nesta camada que se faz o controle da conversação entre as aplicações intercomunicadas da rede.

A camada de transporte utiliza dois protocolos: o TCP e o UDP. Esta camada reúne os protocolos que realizam as funções de transporte de dados fim-a-fim, ou seja, considerando apenas a origem e o destino da comunicação, sem se preocupar com os elementos intermediários. A camada de transporte possui dois protocolos que são o UDP (User Datagram Protocol) e TCP (Transmission Control Protocol). O protocolo UDP realiza apenas a multiplexação para que várias aplicações possam acessar o sistema de comunicação de forma coerente. É conhecido como serviço não orientado a conexão. O protocolo TCP realiza, além da multiplexação, uma série de funções para tornar a comunicação entre origem e destino mais confiável. São responsabilidades do protocolo TCP: o controle de fluxo, o controle de erro, a sequenciação e a multiplexação de mensagens. É conhecido como orientado à conexão.

A camada de transporte oferece para o nível de aplicação um conjunto de funções e procedimentos para acesso ao sistema de comunicação de modo a permitir a criação e a utilização de aplicações de forma independente da implementação. Desta forma, as interfaces socket ou TLI (ambiente Unix) e Winsock (ambiente Windows) fornecem um conjunto de funções-padrão para permitir que as aplicações possam ser desenvolvidas independentemente do sistema operacional no qual rodarão.

O acesso das aplicações à camada de transporte é feito através de portas que recebem um número inteiro para cada tipo de aplicação, podendo também tais portas serem criadas ao passo em que novas necessidades vão surgindo com o desenvolvimento de novas aplicações.

A maneira como a camada de transporte transmite dados das várias aplicações simultâneas é por intermédio da multiplexação, onde várias mensagens são repassadas para a camada de rede (especificamente ao protocolo IP) que se encarregará de empacotá-las e mandar para uma ou mais interface de rede. Chegando ao destinatário o protocolo IP repassa para a camada de transporte que demultiplexa para as portas (aplicações) específicas.

5.1. TCP (*Transmission Control Protocol*)

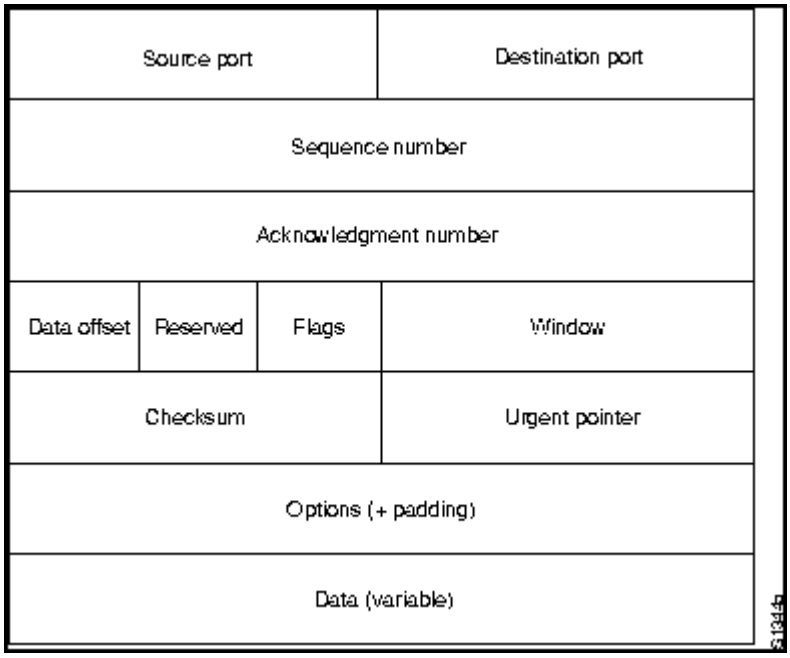
É o protocolo TCP que faz a comunicação fim-a-fim da rede. É orientado à conexão e altamente confiável independente da qualidade de serviços das sub-redes que servem de caminho. Para a confiabilidade de transmissão, garante a entrega das informações na seqüência em que lhe foi fornecida, sem perda nem duplicação.

Principais funções:

a. transferência de dados — Através de mensagens de tamanho variável em *full-duplex*;
b. Transferência de dados urgentes — Informações de controle, por exemplo;
c. estabelecimento e liberação de conexão — Antes e depois das transferências de dados, através de um mecanismo chamado *three-way-handshake*;
d. multiplexação — As mensagens de cada aplicação simultânea são multiplexadas para repasse ao IP. Ao chegar ao destino, o TCP demultiplexa as mensagens para as aplicações destinatárias;
e. segmentação — Quando o tamanho do pacote IP não suporta o tamanho do dado a ser transmitido, o TCP segmenta (mantendo a ordem) para posterior remontagem na máquina destinatária;

f. controle do fluxo — Através de um sistema de buferização denominada janela deslizante, o TCP envia uma série de pacotes sem aguardar o reconhecimento de cada um deles. Na medida em que recebe o reconhecimento de cada bloco enviado, atualiza o buffer (caso reconhecimento positivo) ou reenvia (caso reconhecimento negativo ou não reconhecimento após um timeout);
g. controle de erros — Além da numeração dos segmentos transmitidos, vai junto com o header uma soma verificadora dos dados transmitidos (*checksum*), assim o destinatário verifica a soma com o cálculo dos dados recebidos);
h. precedência e segurança — Os níveis de segurança e precedência são utilizados para tratamento de dados durante a transmissão.

A Figura abaixo mostra o formato do datagrama TCP.

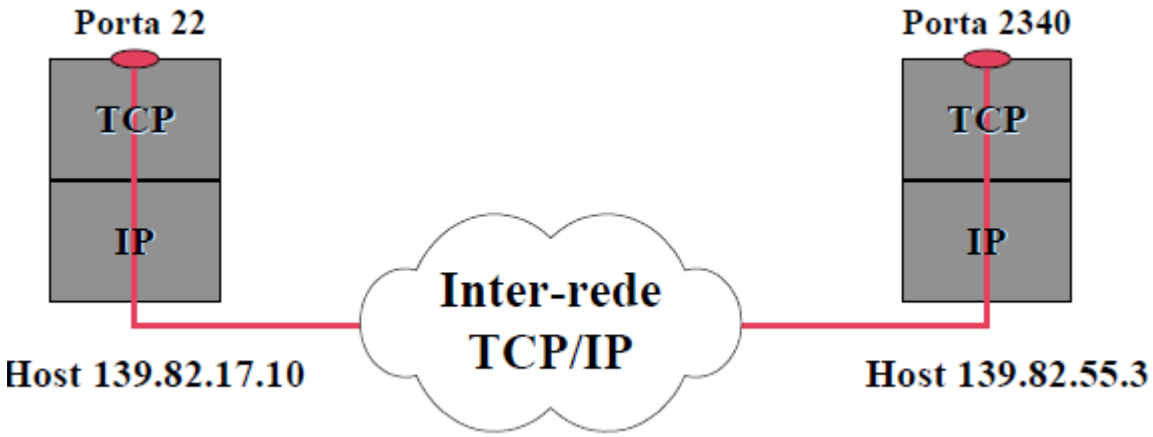


O protocolo TCP trabalha no mesmo nível que o protocolo UDP, mas oferece serviços mais complexos, que incluem controle de erros e fluxo, serviço com conexão e envio de fluxo de dados. TCP utiliza o mesmo conceito de porta de UDP. Para TCP, uma conexão é formada pelo par (End. IP. Origem, Porta Origem) e (End. IP Destino, Porta Destino).

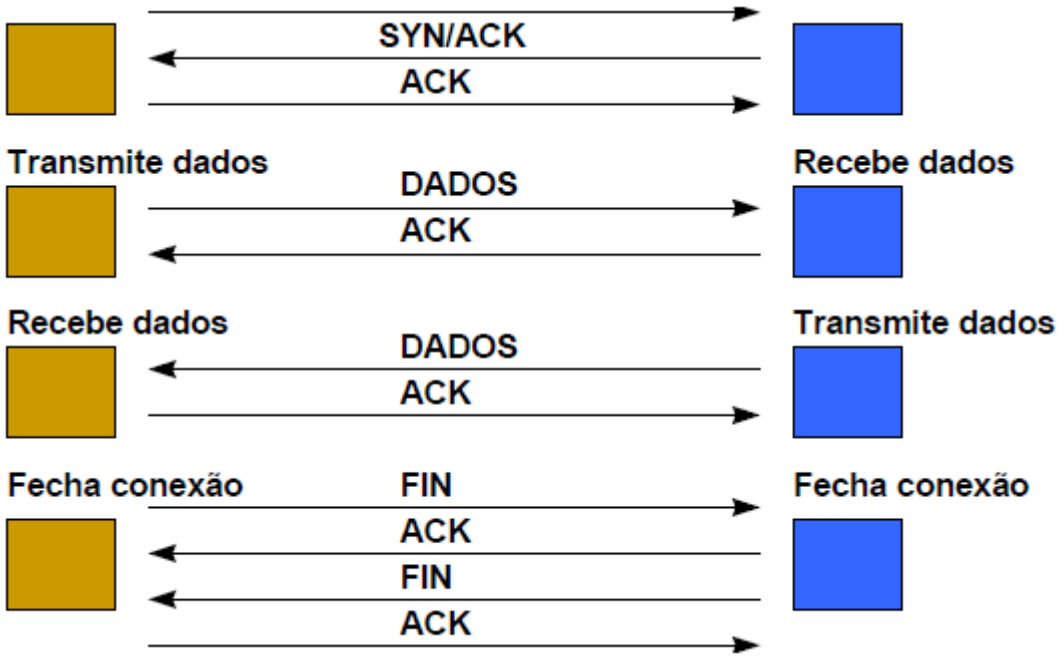
O protocolo TCP oferece as seguintes características:

- Controle de Fluxo e Erro fim-a-fim
- Serviço confiável de transferência de dados
- Comunicação full-duplex fim-a-fim
- A aplicação basta enviar um fluxo de bytes
- Desassociação entre quantidade. de dados enviados pela aplicação e pela camada TCP
- Ordenação de mensagens
- Multiplexação de IP, através de várias portas

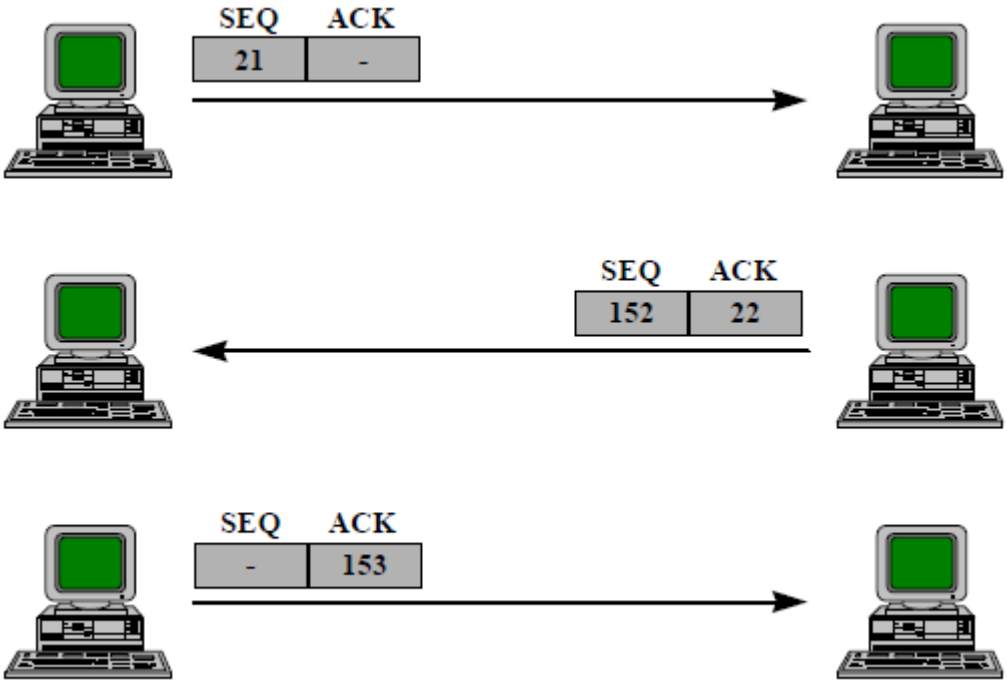
A conexão TCP é ilustrada na figura abaixo:



Uma conexão TCP é formada por três fases: o estabelecimento de conexão, a troca de dados e o finalização da conexão, conforme ilustrado na figura abaixo:



A fase inicial de estabelecimento de conexão é formada de três mensagens, formando o *three-wayhanshaking*, conforme a figura abaixo:



O TCP é orientado a conexão. Um protocolo de transporte orientado a conexão, primeiro, antes de iniciar a transferência de dados, deve estabelecer uma conexão virtual entre origem e destino dos dados. Todos os segmentos pertencentes a uma mensagem são então enviados através dessa conexão virtual. Usar uma única conexão virtual para uma mensagem inteira facilita o processo de confirmação, bem como a retransmissão de segmentos perdidos ou corrompidos.

Poder-se-ia perguntar como o TCP, que usa os serviços do IP, um protocolo sem conexão, pode ser orientado a conexão. O ponto é que uma conexão TCP é virtual, não física. O TCP opera em um nível mais alto. O TCP usa os serviços do IP para transmitir segmentos individuais ao receptor, porém ele controla a conexão em si. Se um segmento for perdido ou corrompido, ele será retransmitido. Diferentemente do TCP, o IP toma conhecimento dessa retransmissão. Se um segmento vier fora de ordem, o TCP o retém até que o segmento faltante chegue; o IP não implementa reordenamento. No TCP, uma transmissão orientada a conexão requer três fases: estabelecimento da conexão, transferência de dados e encerramento da conexão.

Estabelecimento e encerramento da Conexão

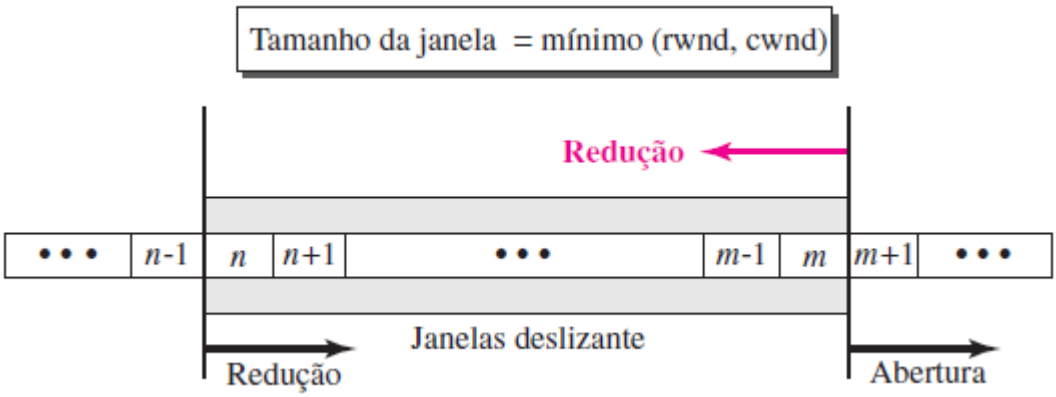
O TCP transmite dados no modo full-duplex. Quando dois processos TCPs em duas máquinas estão conectados, eles estão aptos a transmitir segmentos entre si, simultaneamente. Isso implica que cada parte deve inicializar a comunicação e obter a aprovação da outra parte antes que quaisquer dados possam ser transferidos. Após o estabelecimento de uma conexão, pode-se iniciar a **transferência de dados** bidirecional. O cliente e o servidor podem transmitir tanto dados como confirmações.

O TCP transmissor usa buffers para armazenar o fluxo de dados proveniente do programa de aplicação transmissor. O TCP transmissor pode configurar um tamanho máximo para um segmento. O TCP receptor também coloca os dados em buffers quando eles chegam por completo e, assim, os entrega para o programa de aplicação assim que estiver pronto ou quando for mais conveniente para o TCP receptor. Esse tipo de flexibilidade aumenta a eficiência da transmissão de dados.

Qualquer uma das duas partes envolvidas na troca de dados (cliente ou servidor) pode encerrar uma conexão, embora esta tenha sido, normalmente, iniciada pelo cliente.

Controle de Fluxo

O TCP utiliza a técnica de janela deslizante (*slidding window*) para implementar controle de fluxo.
A Figura abaixo mostra um exemplo de janela deslizante no TCP. A janela abrange parte do buffer, que contém os bytes recebidos do processo. Os bytes dentro da janela são bytes que podem estar em trânsito; eles podem ser enviados sem se preocupar com a confirmação. A janela imaginária possui duas paredes: uma à direita e outra à esquerda.



Uma janela pode ser *aberta, fechada* ou *reduzida*. Essas três atividades, estão sob controle do receptor (e dependem do nível de congestionamento na rede), não do emissor. O emissor tem de obedecer às ordens do receptor. Abrir uma janela significa deslocar a parede direita mais para a direita. Isso permite um número maior de bytes novos no buffer, candidatos a serem transmitidos. Fechar uma janela significa deslocar a parede da esquerda mais para a direita. Isso significa que alguns bytes foram confirmados e o emissor não precisa mais se preocupar com eles. Reduzir a janela significa deslocar para a esquerda a janela da direita. Isso é veementemente desencorajado e não permitido em algumas implementações, pois significa renunciar à

elegibilidade de alguns bytes para transmissão. Este é um problema, caso o emissor já tenha enviado esses bytes. Note que a parede da esquerda não pode se mover para a esquerda, pois isso renunciaria parte das confirmações transmitidas anteriormente.

Uma janela deslizante é usada para implementar maior eficiência à transmissão, bem como controle de fluxo de dados, de modo que o destino não fique sobrecarregado com dados. As janelas deslizantes no TCP são orientadas a bytes.

O tamanho de uma janela em uma conexão virtual é determinado pelo menor número entre dois valores possíveis: *janela receptora* (*rwnd*) e *janela de congestionamento* (*cwnd*). A *janela receptora* é o valor anunciado pelo lado oposto em um segmento contendo confirmação. Este é o número de bytes que o outro lado pode aceitar antes de seu buffer estourar e os dados serem descartados. A janela de congestionamento é um valor determinado pela rede para evitar congestionamento.

Exemplo 1
Qual é o valor da janela receptora (*rwnd*) para o host A, se o receptor, o host B, tiver um tamanho de buffer igual a 5.000 bytes e 1.000 bytes de dados recebidos e não processados?
Solução
O valor de *rwnd* = 5.000 – 1.000 = 4.000. O host B pode receber até 4.000 bytes de dados antes de estourar seu buffer. O host B anuncia esse valor no próximo segmento transmitido para A.

Exemplo 2
Qual o tamanho da janela para o host A se o valor de *rwnd* for 3.000 bytes e o valor de *cwnd* for 3.500 bytes?
Solução
O tamanho da janela é o menor entre os valores de *rwnd* e *cwnd*, que é 3.000 bytes.

Controle de Erros

O TCP é um protocolo de transporte confiável. Isso significa que um programa de aplicação, que entrega um fluxo de dados para o TCP, depende do TCP para entregar em ordem o fluxo inteiro para o programa de aplicação na outra ponta, sem erros, e sem qualquer informação perdida ou duplicada.

O TCP fornece confiabilidade implementado controle de erros sobre os dados. O controle de erros inclui mecanismos para detecção de segmentos corrompidos, perdidos ou fora de ordem e segmentos duplicados. O controle de erros também inclui um mecanismo para correção de erros após eles serem detectados. A detecção e a correção de erros no TCP são obtidas por meio do uso de três ferramentas simples: checksum, confirmação de recebimento e time-out.

Checksum
Cada segmento inclui um campo de checksum que é usado para validar a existência de um segmento corrompido. Se o segmento estiver corrompido, ele será descartado pelo TCP de destino e considerado como perdido. O TCP usa o campo de checksum de 16 bits, que é obrigatória em todos os segmentos.

Confirmação
O TCP usa confirmações para validar o recebimento do segmento de dados. Um segmento de controle que não transporta dados, mas que usa um número de sequência, também deve ser confirmado. Um segmento ACK jamais necessita de confirmação.

Um segmento ACK não consome números de sequência e não podem ser confirmados.

Retransmissão
O cerne do controle de erros é a retransmissão de segmentos. Quando um segmento estiver corrompido, perdido ou com atraso, ele é retransmitido. Em implementações modernas, um segmento é retransmitido em

duas ocasiões: quando o tempo do **timer de retransmissão** se esgota ou quando o emissor recebe três ACKs duplicados.

Nas implementações modernas, ocorre retransmissão caso o timer de retransmissão expire ou se tiverem chegado três segmentos ACK duplicados.

Observe que não existe retransmissão para segmentos que não consomem números de seqüência.
Em particular, não existe retransmissão para um segmento ACK.

Não é ativado nenhum timer de retransmissão para um segmento ACK.

Retransmissão Após RTO Uma implementação recente do TCP mantém um timer **RTO (*Retransmission Time-Out*)** para todos os segmentos pendentes (transmitidos, mas não confirmados). Quando vence o time-out, o primeiro segmento pendente é retransmitido, muito embora a falta de um ACK recebido possa ser devido a um segmento com atraso, um ACK com atraso ou uma confirmação perdida. Note que não existe um timer ativo para um segmento que transporta apenas confirmação, significando que nenhum segmento desse tipo poderá ser reenviado. O valor do RTO é dinâmico no TCP e é atualizado tomando-se como base o **RTT (*Round-Trip Time*, em inglês, tempo de ida e volta)** do segmento. RTT é o tempo necessário para um segmento atingir o destino e uma confirmação ser recebida.

Retransmissão Após Três Segmentos ACK Duplicados A regra anterior sobre a retransmissão de um segmento é suficiente se o valor de RTO não for muito grande. Algumas vezes, porém, um segmento é perdido e o receptor recebe um número tão grande de segmentos fora de ordem a ponto deles não poderem ser salvos (o tamanho do buffer é limitado). Para amenizar essa situação, a maioria das implementações atuais segue a regra dos três ACKs duplicados e retransmite imediatamente um segmento faltante. Esse recurso é conhecido como **retransmissão rápida**.

Segmentos Fora de Ordem
Quando um segmento estiver atrasado, perdido ou tiver sido descartado, os segmentos após este, chegarão fora de ordem. Originalmente, o TCP foi desenvolvido para descartar todos os segmentos fora de ordem, resultando na retransmissão de todos os segmentos faltantes e dos segmentos seguintes. Hoje em dia, a maioria das implementações de TCP não descarta segmentos fora de ordem. Elas os armazenam, temporariamente, em um buffer e colocam um *flag* indicando como segmentos fora de ordem até a chegada dos segmentos faltantes. Note, entretanto, que os segmentos fora de ordem não são entregues para o processo receptor. O TCP garante que os dados são entregues, em ordem, para o processo receptor.

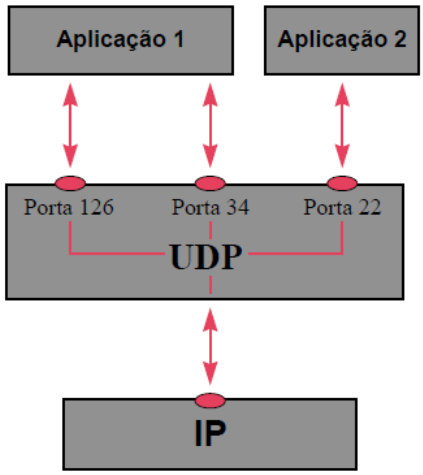
Existe a possibilidade de os dados chegarem fora de ordem e serem armazenados temporariamente pelo TCP receptor, porém o TCP garante que nenhum segmento fora de ordem será entregue ao processo receptor.

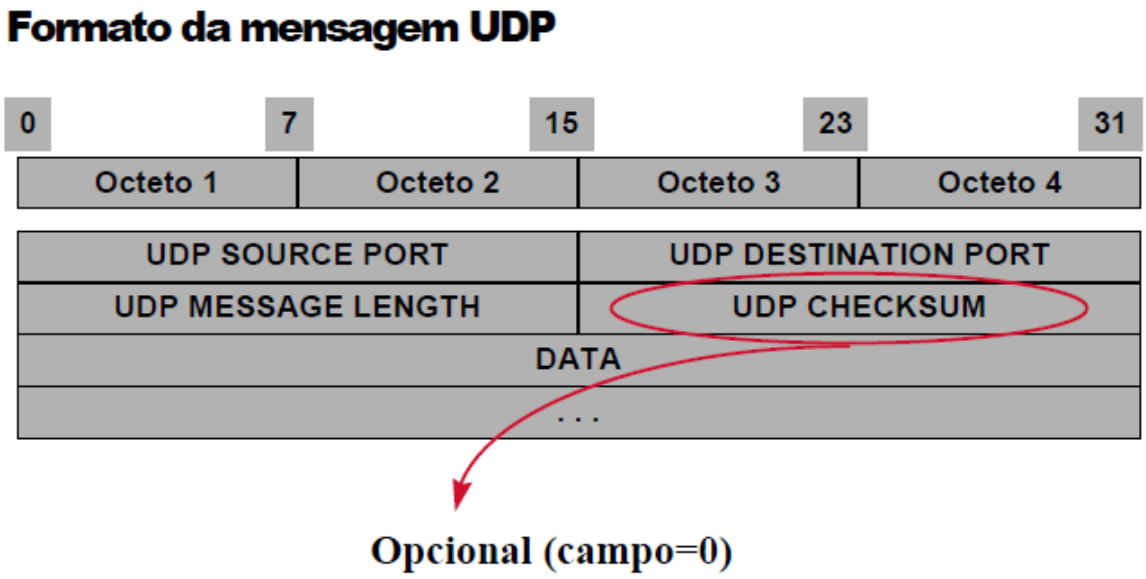
5.2. UDP

Se a confiabilidade não é essencial, o UDP (*User Datagram Protocol*), um complemento do TCP, oferece um serviço de transmissão de dados sem conexão que não garante nem a entrega nem a correta sequência dos pacotes enviados (bem parecido com o IP). *Checksums* no UDP são opcionais, oferecendo assim uma maneira de se trocar dados em uma rede altamente confiável sem consumir desnecessariamente recursos da rede
O protocolo UDP fornece uma forma simples de acesso ao sistema de comunicação, provendo um serviço sem conexão, sem confiabilidade e sem correção de erros. A principal função do nível de transporte implementada em UDP é a capacidade de multiplexação de acesso ao sistema de comunicação. Esta função permite que vários processos ou programas executando em um computador possam acessar o sistema de comunicação e o tráfego de dados respectivo a cada um deles seja corretamente identificado, separado e utilize buffers individuais.

Um processo é o programa que implementa uma aplicação do sistema operacional, e que pode ser uma aplicação do nível de aplicação TCP/IP. A forma de identificação de um ponto de acesso de serviço (SAP) do modelo OSI é a porta de protocolo em TCP/IP. A porta é a unidade que permite identificar o tráfego de dados destinado a diversas aplicações. A identificação única de um processo acessando os serviços TCP/IP é, então, o endereço IP da máquina e a porta (ou portas) usadas pela aplicação. Cada processo pode utilizar mais de uma porta simultaneamente, mas uma porta só pode ser utilizada por uma aplicação em um dado momento.
Uma aplicação que deseje utilizar os serviços de comunicação deverá requisitar uma ou mais portas para realizar a comunicação. A mesma porta usada por uma aplicação pode ser usada por outra, desde que a primeira tenha terminado de utilizá-la. A forma de utilização de portas mostra uma distinção entre a parte cliente e a parte servidora de uma aplicação TCP/IP.
O programa cliente pode utilizar um número de porta qualquer, já que nenhum programa na rede terá necessidade de enviar uma mensagem para ele. Já uma aplicação servidora deve utilizar um número de porta bem-conhecido (Well-known ports) de modo que um cliente qualquer, querendo utilizar os serviços do servidor, tenha que saber apenas o endereço IP da máquina onde este está executando.
Se não houvesse a utilização de um número de porta bem conhecido, a arquitetura TCP/IP deveria possuir um mecanismo de diretório para que um cliente pudesse descobrir o número da porta associado ao servidor. Para evitar este passo intermediário, utiliza-se números de porta bem conhecidos e o cliente já possui pré programado em seu código o número de porta a ser utilizado. Os números de porta de 1 a 1023 são números bem-conhecidos para serviços (aplicações) atribuídos pela IANA (Internet Assigned Numbers Authority). Os números de 1024 a 65535 podem ser atribuídos para outros serviços e são geralmente utilizados pelas programas-cliente de um protocolo (que podem utilizar um número de porta qualquer). Este conjunto de números tem ainda a atribuição de alguns serviços de forma não oficial, já que os primeiros 1024 números não conseguem comportar todos os protocolos TCP/IP existentes.

A figura abaixo ilustra a multiplexação/demultiplexação realizada pelo protocolo UDP, camada de transporte:





6. Camada de Aplicação

É formada pelos protocolos utilizados pelas diversas aplicações do modelo TCP/IP. Esta camada não possui um padrão comum. O padrão estabelece-se para cada aplicação. Isto é, o FTP possui seu próprio protocolo, o TELNET possui o seu próprio, bem como o SNMP, GOPHER, DNS, etc.

É na camada de aplicação 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. Para cada aplicação existe uma porta predeterminada.

6.1. Aplicações TCP/IP

As aplicações, no modelo TCP/IP, não possuem uma padronização comum. Cada uma possui um RFC próprio. O endereçamento das aplicações é feito através de portas (chamadas padronizadas a serviços dos protocolos TCP e UDP), por onde são passadas as mensagens.

Como já mencionado, é na camada de Aplicação que se trata a compatibilidade entre os diversos formatos representados pelos variados tipos de estações da rede.

A comunicação entre as máquinas da rede é possibilitada através de primitivas de acesso das camadas UDP e TCP. Antes de iniciar o estabelecimento da conexão, são executadas as primitivas *sockets*, que cria um ponto terminal de comunicação, e *bind* (ligar) que registra o endereço da aplicação (número da porta). Para estabelecer a conexão (com o protocolo TCP), a aplicação servidora executa a primitiva *listen* enquanto que a cliente executa *connect*. A aplicação servidora usa o *accept* para receber e estabelecer a conexão. Já o UDP, como não é orientado à conexão, logo após o *socket* e *o bind*, utiliza as primitivas *sendto* e *recvfrom*.

Algumas as aplicações TCP/IP são:

1.TELNET (Terminal Virtual)

É um protocolo que permite a operação em um sistema remoto através de uma sessão de terminal. Com isso, a aplicação servidora recebe as teclas acionadas no terminal remoto como se fosse local. Utiliza a porta 23 do TCP.

O TELNET oferece três serviços: Definição de um terminal virtual de rede, Negociação de opções (modo de operação, eco, etc.) e Transferência de dados.

2. FTP (*File Transfer Protocol*)

Provê serviços de transferencia, renomear e eliminação de arquivos, além da criação, modificação e exclusão de diretórios. Para sua operação, são mantidas duas conexões: uma de dados e outra de controle. Não implementa segurança, o que deixa para o TCP, exceto as requisições de senhas de acesso a determinados arquivos (ou servidores FTP).

As transferências de arquivos podem ser no modo TEXTO, onde há conversões de codificação para o sistema destinatário, e o modo BINÁRIO, onde não há nenhuma conversão e todos os bytes são transferidos como estão.

3. SNMP (*Simple Network Management Protocol*)

É utilizado para trafegar as informações de controle da rede. De acordo com o sistema de gerenciamento da arquitetura TCP/IP, existem o agente e o gerente que coletam e processam, respectivamente, dados sobre erros, problemas, violação de protocolos, dentre outros.

Na rede existe uma base de dados denominada MIB (*Management Information Base*) onde são guardadas informações sobre *hosts*, *gateways*, interfaces individuais de rede, tradução de endereços, e softwares relativos ao IP, ICMP, TCP, UDP, etc. Através do SNMP pode-se acessar os valores dessas variáveis, receber informações sobre problemas na rede, armazenar valores, todos através da base do MIB.

4. DNS (*Domain Name System*)

O DNS é um mecanismo para gerenciamento de domínios em forma de árvore. Tudo começa com a padronização da nomenclatura onde cada nó da arvore é separado no nome por pontos. No nível mais alto podemos ter: COM para organizações comerciais, EDU para instituições educacionais, GOV para instituições governamentais, MIL para grupos militares, ORG para outras organizações. Assim, o sistema de resolução de nomes é baseado em uma estrutura de domínios. Geralmente vários servidores participam do processo de resolução de nomes, sendo que cada um é responsavel por resolver os nomes em um determinado domínio.
O DNS possui um algoritmo confiável e eficiente para tradução de mapeamento de nomes e endereços

5. SMTP (*Simple Mail Tansfer Protocol*)

Implementa o sistema de correio eletrônico da Internet, operando não orientado à conexão, provê serviços de envio e recepção de mensagens do usuário. Tais mensagens são armazenadas num servidor de correio eletrônico onde o usuário destinatário está cadastrado, até que este solicite-a, quando são apagadas da área de transferência do sistema originador.

O SMTP divide a mensagem em duas partes: corpo e cabeçalho que são separados por uma linha em branco. No cabeçalho existem uma seqüência de linhas que identificam o emissor, o destinatário, o assunto, e algumas outras informações opcionais.

7. Roteamento

É o processo de escolha do caminho pelo qual o pacote deve chegar à estação destinatária. O roteamento pode ser direto ou indireto.

1.Roteamento Direto

O roteamento direto ocorre quando a estação destinatária do datagrama está na mesma sub-rede física da estação origem. A checagem é feita comparando o endereço IP do emissor e do destinatário constantes no datagrama IP. Nesse caso o conteúdo do datagrama recebe o endereço físico da estação e é enviado diretamente pela mesma sub-rede.

2.Roteamento Indireto

No caso do roteamento indireto, o emissor deve enviar para o *gateway* o datagrama com o endereço IP do destinatário. O *gateway* verificará se o destinatário pertence a uma das sub-redes a ele conectadas, e em caso positivo envia o pacote diretamente para a estação. Caso o *gateway* não localize o destinatário como um membro de uma das sub-redes a ele conectadas, ele envia o pacote para outro *gateway* (de acordo com sua tabela de roteamento), que verificará o mesmo, e assim por diante até encontrar o destinatário ou terminar o tempo de vida do pacote.

3. Algorítmos de Roteamento

São as formas como os gateways localizam as diversas redes e estações. Podem ser: roteamento *Vector-Distance*; Roteamento *Link-State (shortest path first*).

1.Protocolos de Roteamento

Os protocolos de roteamento padronizam a forma como os gateways trocam informações necessárias à execução dos algoritmos de roteamento.

EGP (*Exterior Gateway Protocol*)

Não está vinculado a nenhum algoritmo de roteamento, isto significa que os gateways que se comunicam não necessitam rodar o mesmo algoritmo. Define as informações a serem trocadas entre *Gateways* Exteriores.

É elaborado para uma rede de sistemas autônomos numa topologia em árvore. As mensagens são associadas a cada sistema autônomo através de uma identificação no header da mensagem do EGP. Estas mensagens só trafegam em gateways vizinhos.

Dois gateways tornam-se vizinhos quando trocam mensagens de Aquisição de Vizinho. Após isso, verificam o estado do vizinho através da mensagem de Disponibilidade e através da mensagem Alcance identificam quais redes podem ser acessadas a partir do vizinho.

RIP (*Routing Information Protocol*)

Desenvolvido na Universidade de Berkeley - California, permite a troca de informações com o algoritmo Vector-Distance em uma sub-rede dotada de difusão de mensagens.
Um gateway executando RIP no modo ativo envia informações a cada 30 segundos ou quando solicitado. As mensagens contém informações de todas as tabela de roteamento do gateway. Estas informações são: O endereço IP da sub-rede e a distância do gateway (quantidade de *gateways*). As estações e gateways que recebem as mensagens atualizam sua tabela de acordo com o algoritmo *vector-distance*.

OSPF

Foi desenvolvido por um grupo de trabalho da *Internet Engineering Task Force*, para roteamento de grandes redes. Utiliza o algoritmo de roteamento SPF e possui várias vantagens:

a. roteamento de acordo com o tipo de serviço;
b. balanceamento de carga entre rotas do mesmo tamanho;
c. Definição de rotas específicas para máquinas e redes;

d. modularização do SA, através da criação de áreas que contém gateways e redes. A topologia de tais áreas são conhecidas apenas nesta área.
e. definição de uma topologia de rede virtual que abstraia detalhes da rede real;
f. divulgação de mensagens recebidas de *Gateways* Exteriores.

Quando gateways OSPF são inicializados, eles verificam junto com os *gateways* vizinhos, quem será o *gateway* mestre. O Gateway mestra será encarregado da notificação de informações de roteamento para todos os gateways da sub-rede. Como apenas o *gateway* mestre envia informações, o trafego é reduzido consideravelmente.