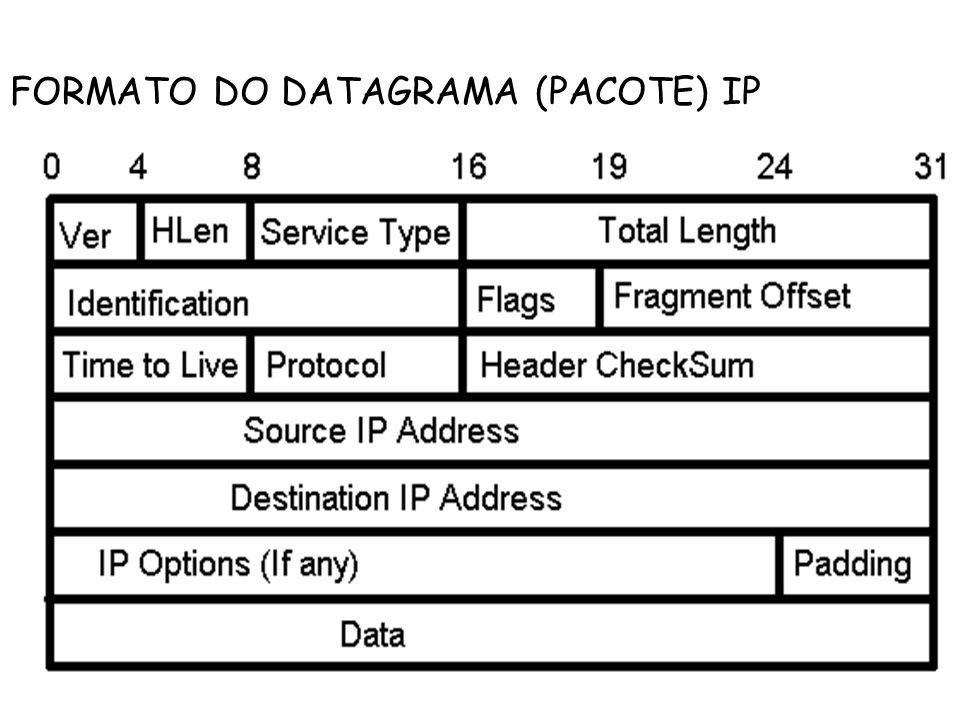
Datagrama ip



VERS: versão do protocolo IP que foi usada para criar o datagrama (4bits)

HLEN: comprimento do cabeçalho, medido em palavras de 32 bits (4 bits)

TOTAL-LENGTH: este campo proporciona o comprimento do datagrama medido em bytes, incluindo cabeçalho e dados.

SERVICE-TYPE: este campo especifica como o datagrama poderia ser manejado e dividido em cinco subcomandos

IDENTIFICATION, FLAGS e FRAGMENTS: estes três campos controlam a fragmentação e a união dos datagramas. O campo de identificação contém um único inteiro que identifica o datagrama, é um campo muito importante porque quando um gateway fragmenta um datagrama, ele copia a maioria dos campos do cabeçalho do datagrama em cada fragmento, então a identificação também deve ser copiada, com o propósito de que o destino saiba quais fragmentos pertencem a quais datagramas. Cada fragmento tem o mesmo formato que um datagrama completo.

FRAGMENT OFFSET: especifica o início do datagrama original dos dados que estão sendo transportados no fragmento. É medido em unidades de 8 bytes.

FLAG: controla a fragmentação.

TTL(Time To Live): especifica o tempo em segundos que o datagrama está permitido a permanecer no sistema Internet. Gateways e hosts que processam o datagrama devem decrementar o campo TTL cada vez que um datagrama passa por eles e devem removê-lo quando seu tempo expirar.

PROTOCOL: especifica qual protocolo de alto nível foi usado para criar a mensagem que está sendo transportada na área de dados do datagrama.

HEADER-CHECKSUM: assegura integridade dos valores do cabeçalho.

SOURCE AND DESTINATION IP ADDRESS: especifica o endereço IP de 32 bits do remetente e receptor.

OPTIONS: é um campo opcional. Este campo varia em comprimento dependendo de quais opções estão sendo usadas. Algumas opções são de um byte, e neste caso este campo é chamado de Option Code , e está dividido em três campos.

Protocolo ARP

A diferença básica entre ARP e RARP é que o ARP, quando fornecido com o endereço lógico do receptor, obtém o endereço físico do receptor, enquanto no RARP, quando fornecido com o endereço físico do host, obtém o endereço lógico do host do servidor.

Protocolo ICMP

O objetivo principal do ICMP é o relatório de erros. Quando dois dispositivos se conectam pela Internet, o ICMP gera erros para compartilhar com o dispositivo de envio, caso algum dos dados não chegue ao destino pretendido. Por exemplo, se um pacote de dados for muito grande para um roteador, o roteador descartará o pacote e enviará uma mensagem ICMP de volta à fonte original dos dados.

Um uso secundário do protocolo ICMP é realizar diagnósticos de rede; os utilitários de terminal comumente usados traceroute e ping operam usando ICMP. O utilitário traceroute é usado para exibir o caminho de roteamento entre dois dispositivos de Internet. O caminho de roteamento é o caminho físico real dos roteadores conectados pelos quais uma solicitação deve passar antes de chegar ao seu destino. A jornada entre um roteador e outro é conhecida como ‘salto’, e um traceroute também informa o tempo necessário para cada salto ao longo do caminho. Isso pode ser útil para determinar as fontes de atraso da rede.

Protocolo UDP

Benefícios ou vantagens do UDP

A seguir estão os benefícios ou vantagens do UDP:

➨Ele usa tamanho de pacote pequeno com cabeçalho pequeno (8 bytes). Esse menor número de bytes na sobrecarga faz com que o protocolo UDP precise de menos tempo no processamento do pacote e de menos memória.

➨Não requer conexão para ser estabelecida e mantida.

➨Também a ausência de campo de confirmação no UDP o torna mais rápido, pois não é necessário esperar pelo ACK ou manter os dados na memória até que sejam ACKed.

➨Ele usa checksum com todos os pacotes para detecção de erros.

➨Pode ser usado em eventos onde um único pacote de dados precisa ser trocado entre os hosts.

Desvantagens ou desvantagens do UDP

A seguir estão as desvantagens ou desvantagens do UDP:

➨É um protocolo de transporte sem conexão e não confiável. Não há janelas e nenhuma função para garantir que os dados sejam recebidos na mesma ordem em que foram transmitidos.

➨Ele não usa nenhum controle de erro. Portanto, se o UDP detectar qualquer erro no pacote recebido, ele o descartará silenciosamente.

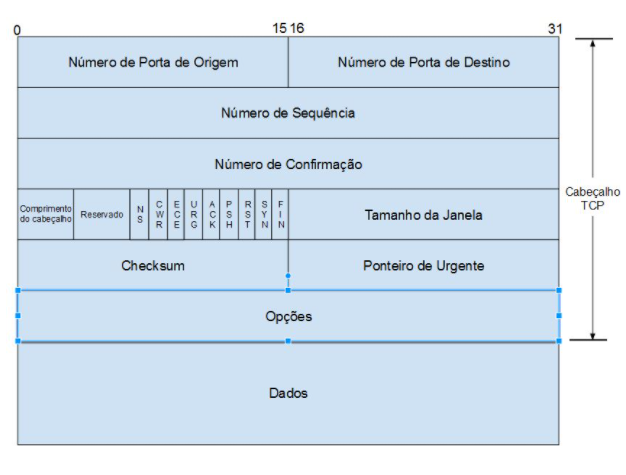
➨Não há controle de congestionamento. Conseqüentemente, um grande número de usuários transmitindo muitos dados via UDP pode causar congestionamento e ninguém pode fazer nada a respeito.

➨Não há controle de fluxo e nenhum reconhecimento de dados recebidos.

➨Apenas a camada de aplicativo trata da recuperação de erros. Conseqüentemente, os aplicativos podem simplesmente recorrer ao usuário para enviar a mensagem novamente.

➨Os roteadores podem ser descuidados com UDP. Eles não retransmitem um datagrama UDP após a colisão e geralmente descartam os pacotes UDP antes dos pacotes TCP.

Protocolo TCP



* **Número de Porta de Origem**: número que identifica a aplicação (processo) que envia os dados
* **Número de Porta de Destino**: número que identifica a aplicação (processo) que deve receber os dados
* **Número de Sequência**: número que identifica o byte em um fluxo de dados do transmissor para o receptor.  
  Se considerarmos o fluxo de bytes em uma direção entre dois hosts, o TCP identifica cada byte com um número de sequência. O número de sequência possui 32 bits de tamanho, com valores possíveis entre 0 e 232 – 1.  
  Quando uma nova conexão é estabelecida, a flag SYN é ativada. Neste caso, o campo número de sequência possui o “número de sequência inicial” (**ISN**, initial sequence number), o qual é escolhido pelo host para esta conexão em particular. O número de sequência do primeiro byte de dados que será enviado será o ISN mais um.  
  Como cada byte que é transmitido é numerado, o número de confirmação (acknowledgment) contém o próximo número de sequência que o host que enviou a confirmação espera receber. Portanto, é o número de sequência mais 1 do último byte de dados que foi recebido com sucesso.
* **Número de Confirmação**: Se a flag ACK estiver ativada o valor deste campo será o próximo número de sequência que o destinatário espera receber. Assim é confirmada a recepção de todos os bytes anteriores, se houverem. A exceção é o primeiro ACK enviado pelos hosts da conexão, que confirma o número de sequência inicial (ISN) de cada um.
* **Comprimento do Cabeçalho** (Data offset): Tamanho do cabeçalho TCP em palavras de 32 bits. O tamanho mínimo é de 5 palavras (20 bytes, sem opções) e o tamanho máximo é de 15 palavras (60 bytes, com opções). Indica o ponto de início dos dados a partir do início do segmento TCP.
* **Reservado** (3 bits): Para uso futuro. Consiste atualmente na sequência 000.
* **Flags**: **ECN** – Explicit Congestion Notification / Notificação Explícita de Congestionamento, adicionados no [RFC 3168](http://www.rfc-editor.org/info/rfc3168) (Cada flag usa 1 bit):
  + NS – Campo opcional adicionado ao ECN para proteger contra dissimulação acidental ou maliciosa de pacotes marcados do transmissor
  + CWR – Congestion Window Reduced, enviada pelo host transmissor para indicar que recebeu um segmento TCP com o flag ECE ativado e respondeu no mecanismo de controle de congestionamento
  + ECE – ECN-Echo. Sua função depende do valor da flag SYN. Se a flag SYN estiver ativada, então o parceiro na conexão TCP é compatível com ECN; se a flag SYN estiver desativada, um pacote com a flag “congestinamento experimentado” no datagrama IP foi recebido durante a transmissão normal
* **Flags: Bits de Controle** (Cada flag usa 1 bit):
  + URG – Quando ativado, indica que o Campo Ponteiro de Urgente é válido e o recurso Transferência de dados com prioridade foi invocado
  + ACK – Bit de Confirmação. Quando ativado, indica que o segmento carrega uma mensagem de confirmação e que o valor do campo Número de Confirmação é válido, e carrega o próximo número de sequência esperado pelo host.
  + PSH – Bit de Push. Quando ativado, indica que o transmissor do segmento usa o recurso TCP Push, o qual requisita que os dados no segmento sejam imediatamente transmitidos à aplicação no dispositivo de destino
  + RST – Bit de Reset. Quando ativado indica que o transmissor encontrou um problema e deseja resetar a conexão.
  + SYN – Bit de Sincronismo. Requisição para sincronizar números de sequência e estabelecer uma conexão entre dois hosts. O campo Número de Sequência contém o número de sequência inicial (ISN) do host que transmite o segmento.
  + FIN – Bit de Finalização. Quando ativado indica que o transmissor do segmento requisita que a conexão seja encerrada.
* **Tamanho da Janela**: O tamanho da janela de recepção indica o número de bytes que o transmissor deste segmento quer aceitar do host de destino, em cada transmissão. Seu valor é também o tamanho da janela de envio do host que recebe este segmento.
* **Checksum**: Usado para verificação de erros do cabeçalho e dos dados transmitidos. Protege o segmento TCP contra erros de transmissão e de entrega dos dados.
* **Ponteiro de Urgente**: Usado em conjunto com o flag URG para transferência de dados com prioridade. Contém o número de sequência do último byte de dados urgentes.
* **Opções**: Campo opcional, que representa informações não cobertas pelos demais campos do cabeçalho TCP. Uma das opções mais comuns é o MSS (Maximum Segment Size). Se o valor do campo de opções não for um múltiplo de 32, então ele será preenchido com zeros (padding) até que seja um valor múltiplo de 32 bits.

Vantagens do TCP

1. Retransmissão de dados

Durante cada handshake, os segmentos TCP são transmitidos do emissor para o receptor. Às vezes, entre o transporte, os segmentos podem se perder ao não chegar ao destino. Devido a isso, uma confirmação será enviada ao remetente do receptor, para que o remetente possa retransmitir o segmento de volta.

2. Controle de congestionamento

Para evitar congestionamentos, o TCP usa uma política de controle de congestionamento separada. Basicamente, o congestionamento acontece quando o remetente envia muitos pacotes de dados em um determinado momento. Para evitar isso, o receptor envia sinais ao remetente para desacelerar o processo ou atrasar a transmissão. A quantidade certa de dados está sendo transmitida para manter a rede saturada.

3. Identificação Única

No TCP, a cada computador da rede foi atribuído um endereço IP exclusivo, tornando-o identificável na rede. Além disso, cada domínio é atribuído com um nome. Portanto, em última análise, o TCP oferece benefícios de serviços de resolução de nomes e endereços.

4. Na entrega do pedido

Sempre que um pacote é enviado, não é possível garantir que estará em ordem quando chegar ao host. O pedido pode se perder no meio. Portanto, antes de chegar ao aplicativo, o TCP executa as etapas necessárias para reorganizá-los em ordem.

5. Detecção de erros

O erro no TCP pode impactar negativamente no desempenho e nos serviços de conectividade. Detectar erros como segmentos corrompidos e ausentes é relativamente fácil no TCP. Geralmente é feito em 3 etapas. Essas são a soma de verificação, retransmissão e confirmação.

Desvantagens do TCP

1. Início lento

O processo de TCP é sempre lento no início. Só depois de um certo período de tempo a velocidade será acelerada. Isso pode ser especialmente desvantajoso ao fazer download de grandes fluxos de dados. Sempre que você não conseguir obter a velocidade total ao tentar baixar arquivos maiores, o processo será lento no início e, eventualmente, a velocidade aumentará até que o processo de download seja concluído.

2. Bloqueios de imagem

Se houver mais de uma imagem presente em uma página da Web e no caso de uma delas se perder. O TCP não será capaz de entregar outras partes do fluxo. O que significa que outras imagens não serão carregadas se houver problemas em uma delas (os dados não serão entregues ao navegador até que todas as imagens sejam carregadas). Esses tipos de problemas semelhantes podem ser especialmente perceptíveis ao assistir a vídeos online. Assim que a conexão for perdida, o TCP será recuperado mesmo se outras partes do vídeo puderem ser assistidas.

3. Resultado da perda do congestionamento

Sempre que houver um congestionamento de dados, o TCP ficará lento para que possa enviar o tráfego em uma taxa constante. Embora isso possa ser benéfico para redes com fio, não é vantajoso no caso de uma conexão sem fio. Mesmo que haja uma queda na conexão em uma rede sem fio, o TCP deve assumir que a conexão não é confiável e continuar enviando tráfego na mesma taxa.

4. Aperto de mão lento

Sempre que uma conexão for estabelecida, o TCP estará realizando um handshake entre o remetente e o destinatário. Em última análise, esse processo pode fazer com que o estabelecimento da conexão fique mais lento. No entanto, isso não será problemático até que haja um uso de dispositivos com alta latência. Geralmente, esses dispositivos podem demorar muito para estabelecer uma conexão TCP.

5. Otimização de rede

Originalmente, uma conexão TCP é otimizada apenas para redes de longa distância (WAN). Não foi projetado para redes pequenas, como Rede de área local (LAN) e Rede de área pessoal (PAN). Portanto, o usuário deve mudar para os serviços WAN para obter o máximo de benefícios dessa conexão.