Capítulo 1

1.1 O que é a internet?

1.1.1 Uma descrição dos componentes da rede

Hospedeiros ou sistemas finais são os computadores de mesa, servidores, ou outros aparelhos conectados na internet que executam aplicações

São conectados por enlaces (links) de comunicação e comutadores (switches) de pacotes

A taxa de transmissão de um enlace é a quantidade de bits por segundo transmitida.

Um sistema final comunica-se com outro por mensagens desmontadas, chamadas de pacotes.

Comutador de pacotes recebe o pacote e encaminha para o dispositivo correto.

Sistemas finais acessam a rede por meio de ISPs (Provedores de internet). Cada ISP é uma rede de comutadores de pacotes e enlaces de comunicação.

Sistemas finais se comunicam por protocolos (TCP, IP, HTTP), um conjunto de regras que permite a comunicação. Em geral, são desenvolvidos pela IETF

1.1.2 Uma descrição de serviço

Aplicações distribuídas envolvem diversos sistemas finais que trocam informações.

Sistemas finais possuem uma API que dita como um programa em um sistema final irá se comunicar com outro

1.1.3 O que é um protocolo?

Conjunto de regras que definem o formato e ordem das mensagens trocadas, bem como as ações realizadas na transmissão e recebimento da mensagem ou evento.

Se um computador acessa uma página web, o computador envia uma mensagem de requisição de conexão para o servidor, que recebe a mensagem e retorna uma

resposta para a conexão, o computador então faz a sua solicitação da página web e o servidor retorna.

1.2 A periferia da internet

Sistemas finais incluem computadores de mesa, servidores, celulares ,etc.

Sistemas finais são chamados de hosts por hospedarem (executarem) aplicações (navegadores, leitores de email, etc).

Sistemas finais podem ser divididos entre clientes (computadores de mesa) e servidores (máquinas com mais processamento e armazenamento).

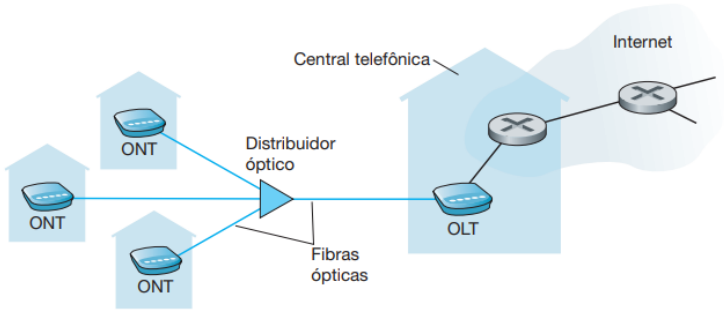
1.2.1 Redes de acesso

Rede de acesso é a rede física que conecta um sistema final a um primeiro roteador

O acesso por uma linha digital de assinante (DSL) é feita a partir do acesso telefônico. Quando uma mensagem é enviada, o sinal digital é transformado em som de alta frequência para transmissão até a operadora

Os dados são divididos em dois canais, um de *downstream* para envio de dados e um *upstream* para recebimento de dados. Um pacote enviado pelo terminal viaja pelo downstream até cada residência, e o pacote enviado por uma residência viaja pelo upstream até o terminal de transmissão.

Hoje temos as redes FTTH, fibras ópticas. A fibra sai da central, é compartilhada para várias residências e é dividida individualmente ao se aproximar das casas.



Cada residência conecta-se a um distribuidor que alcança N casas, depois manda o sinal para a central e ela conecta-se com a internet de fato.

1.2.2 Meios físicos

É por onde os bits são transportados para chegarem ao destino

Meios guiados: Ondas são dirigidas em um meio sólido como a fibra óptica, par trançado, cabo coaxial

Meios não guiados: As ondas se propagam pelo ar, como no wifi e bluetooth

1.3 O núcleo da rede

1.3.1 Comutação de pacotes

Uma mensagem é fragmentada em pedaços chamados de pacotes. Entre a origem e o destino, eles passam por comutadores de pacotes (roteadores e comutadores da camada de enlace).

$Tempo\ de\ transmissão\ = \frac{L\ bits}{R\ bits/s}$ Onde L é o tamanho do pacote e R é a taxa de transmissão do enlace.

Cada sistema final possui um IP, assim ao enviar um pacote, a aplicação coloca no cabeçalho do pacote o IP do destino e o envia. O pacote é recebido em um comutador que lê o IP de destino e, a partir de uma **tabela de encaminhamento**, decide qual o próximo enlace para encaminhar o pacote.

Quando um pacote chega, ele é totalmente recebido antes de ser reenviado, para isso, ele é armazenado em um **buffer**. Ademais, caso haja muitos itens para serem enviados, pode gerar um atraso de fila no **buffer de saída**. Assim, com mais dados entrando do que saindo, o buffer fica cheio e passa a ignorar os pacotes que chegam, gerando uma **perda de pacote**.

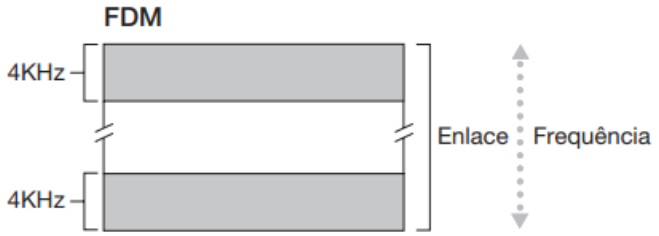
1.3.2 Comutação de circuitos

Os recursos usados ao longo do caminho (buffers, taxa de transmissão) são reservados pelo período de sessão dos sistemas finais, impedindo a ocorrência de filas.

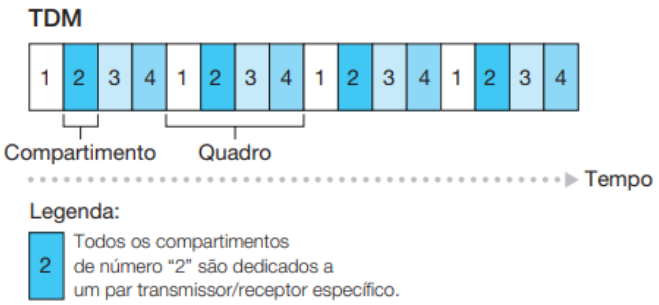
Em uma rede telefônica, antes de enviar uma informação, é criada uma conexão fim a fim forte chamada de circuito. Nela, o caminho do destinatário e remetente mantêm o estado com uma taxa de transmissão constante nos enlaces durante a sessão.

Multiplexação e redes de comutação de circuitos

Multiplexação por divisão de frequência (FDM): Durante todo o período da conexão uma taxa da frequência ficará reservada. Assim, cada conexão usa uma taxa



Multiplexação por divisão de tempo (TDM): Durante um curto período de tempo, pode ser utilizada toda a frequência disponível, depois libera para outra conexão



A comutação de pacotes baseia-se no fato de que os usuários estão ativos somente 10% do tempo. Também utilizam o fato de haver uma baixa probabilidade de muitos usuários estarem ativos ao mesmo tempo, dessa forma, uma rede pode possuir muito mais usuários do que teria se fosse numa comutação por circuito, onde os usuários estariam sempre "ativos".

1.3.3 Uma rede de redes

ISPs acesso (conecta usuários finais a provedores de conteúdos) se comunicam com um ISP global (rede de roteadores e enlaces que se espalha pelo mundo), que cobra do ISP de acesso uma taxa para a conexão.

Porém existem diferentes ISPs globais que precisam estar interconectados para permitir uma conexão geral

Dessa forma, os ISPs locais(conectam usuários finais) se conectam a ISPs regionais (Level 3)(Conectam cidades) se conectam a ISPs de acesso (Level 2)(Alcançam todo o país) que se conectam a ISPs globais(Level 1)(Alcançam todo o mundo e são os que fazem as conexões marítimas).

ISPs de mesmo nível, muitas vezes, se emparelham e conectam diretamente às suas redes para não precisarem pagar por conexões mais altas.

Hoje em dia, provedores de conteúdo, muitas vezes, criam as suas próprias redes para transporte de dados privados entre diferentes servidores ao redor do mundo. Mas para conectar com os usuários finais, contratam ISPs locais quando possível e também ISPs de nível 1 quando não conseguem fazer eles mesmos.

1.4 Atraso, perda e vazão em redes de comutação de pacotes

1.4.1 Uma visão geral de atraso em redes de comutação de pacotes

Um pacote é enviado de um nó para outro por meio de um roteador A, que, quando o recebe, examina o cabeçalho e determina o melhor caminho para o roteador B. Sendo que o pacote só será liberado quando a saída estiver limpa.

Atraso de processamento: Tempo gasto para direcionar um pacote e corrigir erros do caminho

Atraso de fila: Espera para ser transmitido no enlace

Atraso de transmissão: Tempo gasto para empurrar todos os bits do pacote para fora do nó atual. Definido por $Tempo\ de\ transmissão\ = \frac{L\ bits}{R\ bits/s}$ Onde L é o tamanho do pacote e R é a taxa de transmissão do enlace.

Atraso de propagação: O tempo que demora pro bit se propagar no enlace. Sendo que o tempo varia de acordo com o meio de propagação. $Atraso = \frac{comprimento\ do\ enlace}{velocidade\ de\ propagação\ no\ meio}$

$Atraso\ nodal = d_{proc} + d_{fila} + d_{transmissão} + d_{propagação}$

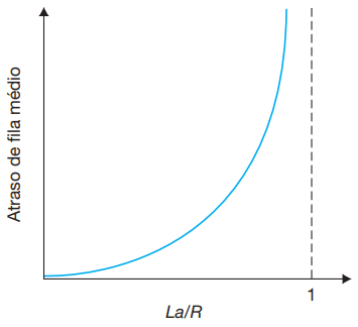
1.4.2 Atraso de fila e perda de pacote

Se dez pacotes chegam em uma fila vazia ao mesmo tempo, o primeiro não sofre atraso, mas o segundo sim e o valor vai aumentando até chegar ao último.

$Intensidade\ do\ tráfego = \frac{L * a}{R}$

Onde 'a' é a taxa (pacotes/segundos) que os pacotes chegam na fila; L é o tamanho dos pacotes em bits e R é a taxa de transmissão em bits/segundo

- La/R > 1: Chega mais do que sai, o atraso é muito grande e tende a crescer, podendo ocasionar em perdas de pacote quando o buffer encher
- La/R ≤ 1: Há um atraso considerável. Se chegar N pacotes ao mesmo tempo, o atraso será de LN/R segundos
- La/R ~= 0: O atraso é pequeno, quase inexistente, e tende a diminuir



1.4.3 Atraso fim a fim

O atraso na vida real é a soma de todos os atrasos nodais.

O programa traceroute permite saber por onde os pacotes passaram e quanto tempo demorou. Ele envia 3 pacotes e, assim que atingem roteadores no meio do caminho, eles retornam uma mensagem para a origem, que consegue identificar o roteador e o tempo que durou.

1.4.4 Vazão nas redes de computadores

Vazão é a taxa na qual os bits são transferidos entre o transmissor e o receptor.

Se o arquivo tem F bits e eva T segundos para receber os F bits, a vazão é $R = \frac{F}{T}$

Se um servidor e um cliente trocam informações, a vazão será o valor mínimo da vazão dos dois, ou seja, min{$R_{Cliente}$, $R_{Servidor}$}. Pode-se adicionar nesta função as vazões dos comutadores de pacotes que estão no meio do caminho, mas seus valores podem ser desconsiderados devido à grande capacidade dos mesmos hoje em dia.

Dessa forma, o arquivo terá uma vazão de F / min{$R_{Cliente}$, $R_{Servidor}$}

1.5 Camadas de protocolo e seus modelos de serviço

1.5.1 Arquitetura em camadas

Camadas de protocolo: Cada camada fornece um serviço para a que está acima, sendo que elas juntas formam uma pilha de protocolos.

Camada de aplicação: Sistemas finais executam aplicações que utilizam de protocolos como HTTP (sites), SMTP (email) e FTP (transferência de arquivos) para a comunicação e troca de mensagens entre os hosts.

Camada de transporte: Transporta mensagens entre cliente e servidor de uma aplicação pelos protocolos TCP (entrega garantida e controle de fluxo para evitar congestionamento) e UDP (ele entrega, mas sem garantia que chegou)

Camada de rede: Encaminha datagramas (pacotes da camada de rede) da origem até o destino pelo protocolo IP

Camada de enlace: Transporta pacotes entre diferentes nó

Camada física: Movimento os bits individuais de um nó para o outro

Capítulo 2 - Camada de aplicação

2.1 Princípios de aplicações de rede

2.1.1 Arquiteturas de aplicação de rede

Arquitetura cliente servidor: Um sistema final chamado de servidor fica sempre ligado utilizando um endereço de IP fixo e conhecido, recebendo requisições de clientes, outros sistemas finais que podem ter IPs variáveis

Arquitetura P2P: Sistemas finais com IPs dinâmicos chamados piers conectam-se entre si transferindo arquivos. Caso um terceiro queira se conectar, ele pode e a conexão de download de arquivos pode melhorar, e caso saia, pode piorar

2.1.2 Comunicação entre processos

Dois processos (aplicações) comunicam trocando mensagens

Quem inicia a conexão é o cliente e quem espera ser contactado é o servidor

No P2P, quando A solicita algo a B, o A é cliente, e o B é servidor

A troca de mensagens entre diferentes processos através da rede é feita por meio de sockets

Ao enviar um dado, o programador controla as mensagens, mas nada se faz em relação às outras camadas, podendo, no máximo, definir o protocolo de envio (TCP ou UDP) e o tamanho máximo do buffer.

Para enviar a mensagem, precisa-se saber quem receberá (IP do destino com 32 bits) e qual processo irá receber (a porta na qual o processo está escutando)

A mensagem é enviada pelo socket e do outro lado o protocolo da camada de transporte leva a mensagem da rede até o socket do processo destinatário.

2.1.3 Serviços de transporte disponíveis para aplicações

Serviços de um protocolo de comunicação

Transferência confiável de dados: Uma mensagem é enviada para que mandou confirmando o recebimento do pacote

Vazão: Alguns apps podem necessitar de uma vazão mínima constante, sendo estes conhecidos como aplicações sensíveis à largura de banda. Mas há aplicativos que isso não é necessário

Temporização: Garante que um bit chegará no destino em um determinado espaço de tempo

Segurança: Criptografia e integridade dos dados

2.1.4 Serviço de transporte providos pela Internet

TCP:

- Faz o cliente e servidor fazem uma apresentação ao trocarem informações de controle da camada de transporte antes de trocarem mensagens, gerando uma conexão full-duplex, onde ambos enviam e recebem
- O serviço de transporte é confiável e os pacotes chegam sem erro e na ordem correta
- Evita congestionamento da rede ao limitar a capacidade de transmissão

UDP

- Serviço não confiável : Mensagens podem chegar fora de ordem ou não chegar
- Um processo pode congestionar a rede se enviar muitos dados, já que não há controle

Ambos os protocolos não fornecem uma temporização e vazão garantidas

2.2 A web e o HTTP

2.2.1 Descrição geral do HTTP

Uma página web é feita de objetos (arquivos) que podem ser acessados por URLs.

Um HTML com cinco imagens é, na verdade, seis objetos que se referenciam

O cliente HTTP usa o TCP e cria uma conexão (socket) com o servidor. Com a conexão estabelecida, as mensagens são trocadas pelos sockets e entregues via protocolo TCP

O HTTP é um protocolo sem estado, pois não guarda informações dos pedidos anteriores

2.2.2 Conexões persistentes e não persistentes

Conexões não persistentes: Um objeto é enviado por somente uma conexão. Cada requisição gera uma conexão nova

1. Cliente inicia uma conexão TCP com o servidor na porta 80, onde há um socket no cliente e servidor
2. O cliente envia a solicitação de um arquivo por meio do seu socket
3. O servidor recebe a mensagem por meio do seu socket, encontra o arquivo, encapsula e envia
4. Quando o TCP tem certeza que o cliente recebeu a mensagem, a conexão é encerrada
5. O cliente recebe uma resposta e a conexão é encerrada
6. O HTML possui referências para outros arquivos, então os passos 1 a 4 são repetidos para cada imagem

Tempo de viagem ida e volta (round-trip time) é o tempo que um pacote leva para ir do cliente ao servidor e do servidor ao cliente, incluindo todos os atrasos.

A conexão não persistente gasta 2 RTTs, um para abrir a conexão e outro para enviar os dados.

Conexões persistentes: Permite o envio de vários objetos por meio de uma única conexão TCP, que é fechada após passar um certo tempo.

2.2.4 Interação usuário-servidor: cookies

Componentes dos cookies: linha de cabeçalho do cookie na requisição e resposta do HTTP; Arquivo de cookie mantido no sistema final do usuário e gerenciado pelo navegador; Um banco de dados de apoio do site

Uma pessoa acessa um site -> É criado um ID e é adicionado ao banco de dados -> O servidor responde ao navegador e põe no cabeçalho o id -> o navegador adiciona uma linha ao arquivo de cookies -> o arquivo possui o id e servidor que criou

2.2.5 Caches Web (Servidor Proxy)

É um servidor de cache que fica no meio do caminho com o desejado, em geral nos ISPs. Ele armazena cópias das páginas que foram recentemente usadas, se não possuir, então contacta o servidor desejado e obtém os dados e guarda na cache.

A cache atua tanto como servidor (quando retorna uma informação que possui), como cliente (quando solicita algo que não possui)

2.4 Correio eletrônico na internet

Agentes de usuário: Serviços que permitem a leitura e escrita de email

Servidores de correio: Envia as mensagens para os destinatários

O email é escrito em um agente de usuário, o mesmo encaminha para o servidor de correio, que adiciona o email a uma fila e depois a envia para o servidor de correio do destinatário, e ele envia para o agente de usuário correto.

2.4.1 SMTP

É um protocolo de envio de informações (push) entre servidores de correio que utiliza o TCP.

Transfere mensagens de servidores de correios remetentes para servidores de correio destinatários.

Um problema é que somente envia mensagens no formato ASCII de 7 bits, fazendo com que dados multimídias tenham de ser codificados para envio e codificados para leitura.

1. Uma pessoa escreve um email e instrui para quem o agente de usuário deverá enviar.
2. O agente de usuário envia a mensagem para seu servidor de correio através do SMTP, que coloca a mensagem na fila de envios

3. O lado cliente do SMTP vê a mensagem na fila e abre uma conexão para outro servidor SMTP na porta 25 dele. Se não estiver funcionando, tenta mais tarde.
4. É feito um handshaking (Apresentação dos endereços de email do remetente e destinatário) e o cliente SMTP envia a mensagem pelo TCP
5. O lado servidor SMTP recebe a mensagem e coloca na caixa postal do destinatário
6. O usuário usa seu agente de usuário para ler a mensagem

Se o servidor remetente for enviar múltiplas mensagens para um mesmo servidor destinatário, pode ser feita uma única conexão TCP.

2.4.3 Formatos de mensagem de correio

Todo cabeçalho deve ter as informações de From e To, podendo incluir um assunto e outros opcionais, depois há uma separação por uma linha branco e logo em seguida o corpo do email de fato

2.4.4 Protocolos de acesso ao correio

Para que a mensagem saía do servidor de correio e chegue no agente de usuário é preciso usar um protocolo diferente do SMTP (pois este somente envia dados)

POP3: Começa quando o cliente abre uma conexão TCP com o servidor de correio na porta 110. Passa por uma autorização para autenticar o usuário. Depois ocorre uma transação para recuperar a mensagem, podendo marcá-la. A fase de atualização ocorre quando o cliente encerra a sessão, e o servidor apaga as mensagens marcadas

IMAP: Associa cada mensagem a uma pasta. Quando a mensagem chega pela primeira vez ela é associada ao inbox, e o usuário pode transferir as mensagens entre as pastas, mantendo as atualizações de forma remota.

HTTP (Email pela WEB): Permite a visualização dos e-mails pela internet, sem necessitar do POP ou IMAP. Quando uma mensagem é enviada, ela vai para o servidor de correio por meio do HTTP, mas o resto continua por SMTP

2.5 DNS: O SERVIÇO DE DIRETÓRIO DA INTERNET

2.5.1 Serviços fornecidos pelo DNS

Um host pode ser identificado por um nome ou por um IP

O DNS traduz os nomes dos hospedeiros para IPs

O DNS é um banco de dados que possui uma hierarquia em servidores DNS e é também um protocolo da camada de aplicação permitindo que hosts consultem os dados. O DNS é um protocolo UDP que executa na porta 53

1. O navegador extrai o nome do site e passa para o lado cliente da aplicação DNS
2. O cliente DNS pergunta para o banco de dados DNS qual o IP
3. o navegador recebe o endereço e pode abrir uma conexão com esse IP

Além de traduzir nomes, o DNS também tem os seguintes serviços:

- Apelidos (alisaing) de hosts e servidores de email: O nome canônico pode ser mais difícil de ser lembrado, então um nome mais simples leva para o mesmo local
- Distribuição de cargas: Um único nome pode referenciar diferentes servidores espalhados para evitar a sobrecarga de uma máquina

2.5.2 Visão geral do modo de funcionamento do DNS

O DNS possui vários servidores organizados hierarquicamente e distribuídos pelo mundo, com mapeamentos distribuídos pelos servidores

Servidores DNS raiz: Procurado por um servidor local que não consegue resolver o nome

Servidores DNS de Domínio de alto nível (TLD): responsáveis por domínios como com, org, net, edu e gov e para domínios de países

Servidores DNS autorizativos: São servidores de organizações que podem ser acessados publicamente. São mantidos pela instituição ou paga algum provedor.

DNS local: Não pertence necessariamente à hierarquia de servidores. Cada ISP possui o seu. Quando uma consulta é feita, tenta resolver, se não consegue, envia para cima na hierarquia

Uma consulta pode ser feita de duas formas, iterativa e recursiva

Iterativa

- Uma consulta deseja ser feita
- O DNS local é acessado
- O DNS local consulta o DNS raiz que retorna um endereço TLD
- O DNS local consulta o TLD recebido e recebe uma resposta do mesmo
- O DNS local vai para o endereço autorizativo recebido
- O endereço autorizativo vai até a máquina desejada e retorna para o servidor local
- O servidor local volta as informações para o cliente

Recursiva

- Uma consulta deseja ser feita
- O DNS local é conectado
- O DNS local consulta o DNS raiz
- O DNS raiz consulta diretamente o TLD
- O TLD consulta diretamente o DNS autorizativo
- O DNS autorizativo retorna as informações desejadas da máquina
- As respostas vão voltando passo a passo até chegar no cliente inicial

O DNS utiliza o UDP e caso em algum momento uma resposta não seja recebida, seja porque a mensagem ou a resposta foi perdida, então envia para outro servidor ou fala para o cliente que a conexão não pode ser feita.

Quando um servidor DNS recebe uma resposta DNS, ele pode adicionar o mapeamento em sua memória local. Assim, uma futura requisição para o mesmo endereço irá ser mais rápida. A informação fica guardada na cache por um certo tempo, chamado de time to live (ttl), em geral, dois dias, o que pode acabar causando desatualizações de informações.

2.5.2 Registros e mensagens DNS

Registros de recursos (RR) são armazenados no BD do DNS e fornecem mapeamento de nome de hosts para IPs, sendo que cada registro possui os campos (Name, Value, Type, TTL)

Capítulo 3 - Camada de transporte

3.1 Introdução e serviços de camada de transporte

Fornece comunicação lógica entre processos em diferentes hosts

A camada de transporte converte as mensagens que recebe de um processo de aplicação remetente em pacotes da camada de transporte, chamados de **segmentos** de camada de transporte, feitos a partir da fragmentação de uma mensagem. Depois a mensagem é encapsulada em um pacote e enviada.

3.1.1 Relação entre as camadas de transporte e de rede

Camada de transporte oferece uma conexão entre processos e a camada de rede oferece uma conexão entre hospedeiros

3.1.2 Visão geral da camada de transporte na internet

O UDP fornece somente a entrega de dados e a verificação de erros

O TCP oferece uma transferência confiável de dados e um controle de congestionamento, fazendo com que várias conexões usem a mesma taxa.

3.2 Multiplexação e demultiplexação

Demultiplexação: A camada recebe um segmento e envia ao socket correto através da porta informada

Multiplexação: Na origem, reúne os dados dos sockets, encapsula-os, segmenta e envia para a camada de rede, que enviará uma tupla com o IP destino e porta destino

As portas 0 a 1023 são para serviços conhecidos e as portas 1024 a 65535 podem ser usadas pelo programa.

O socket UDP é uma tupla com Ip destino e porta destino

O socket do TCP é uma quádrupla com (IP`origem, Porta origem, IP destino e Porta destino)

3.3 Transporte não orientado para conexão: UDP

Realiza a (de)multiplexação e corrigi alguns erros;

Recebe a mensagem da aplicação, anexa a porta de origem e destino e passa a informação para a próxima camada.

Se os dados chegaram corretamente, o UDP irá usar o número da porta para entregar os dados corretamente.

O UDP não possui a etapa de apresentação

Características do UDP:

- Assim que recebe uma informação, ele já envia para o destino ignorando possíveis congestionamentos da rede e ignorando a verificação se chegou ou não.
- O UDP envia uma mensagem sem antes se apresentar ao destinatário
- Não mantém estados de conexão e não monitora congestionamento.
- O cabeçalho ocupa menos tamanho

Por não possuir um controle de tráfego, altas taxas de dados podem acabar gerando congestionamento e um aumento na perda de pacotes

3.3.1 Estrutura do segmento UDP

O cabeçalho possui quatro campos de 2 bytes cada, sendo eles o número da porta de origem e destino, comprimento da mensagem e soma de verificação

A verificação de erros é feita como forma de redundância para as outras camadas, onde é verificado se a soma dos bits deve ser a mesma passada no cabeçalho.

3.4 Princípios da transferência confiável de dados

Um canal confiável é onde os dados não sofrem alterações ao sair de um processo para o outro.

3.4.1 Construindo um protocolo de transferência confiável de dados

Transferência confiável de dados sobre um canal perfeitamente confiável: rdt1.0

Considerando que o receptor é completamente confiável.

O remetente e o destinatário possuem diferentes máquinas de estados finitos.

O lado do remetente espera dados da camada superior pelo rdt_sent e depois cria um pacote com o make_pkt, que é enviado pelo udt_send. Depois o processo volta a esperar uma informação da camada de cima.

Do lado do destinatário, quando um pacote chegar, ele é lido pelo rdt_recv extraido pelo extract e enviado para cima pelo deliver_data.

Transferência confiável de dados por um canal com erros de bits: rdt2.0

Em um modelo mais realista, os bit podem ser corrompidos nos componentes físicos.

Protocolos de transferência de dados confiáveis são chamados de **protocolos ARQ (Automatic Repeat reQuest),** onde os dados entendidos corretamente recebem reconhecimentos positivos e, o contrário, reconhecimentos negativos. E possuem três capacitações para manipular erros:

- Detecção de erros: Feita uma soma de verificação semelhante a do UDP
- Realimentação do destinatário: O destinatário envia pacotes de reconhecimento positivo (ACK) e negativo (NAK)
- Retransmissão: Um pacote transmitido com erro é retransmitindo

No lado remetente, após esperar os dados serem passados pela camada superior, um pacote é criado com os dados a serem enviados e a soma de verificação do pacote, para depois enviar o mesmo pelo udt_send(). Depois disso, ficará esperando por uma resposta, seja ela ACK (os dados foram enviados corretamente e volta a esperar outra mensagem) ou NAK (Retransmite o último pacote e fica tentando até receber um ACK), gerando um protocolo pare e espere.

Já no lado do destinatário, há um único estado, espera uma chamada de baixo, quando recebe, verifica se está tudo certo, se sim, envia um ACK, se não envia um NAK.

Para evitar casos onde o ACK ou NAK estejam corrompidos, o remetente envia um número de sequência do pacote atual. O destinatário verifica esse número e determina se o dado é de uma retransmissão.

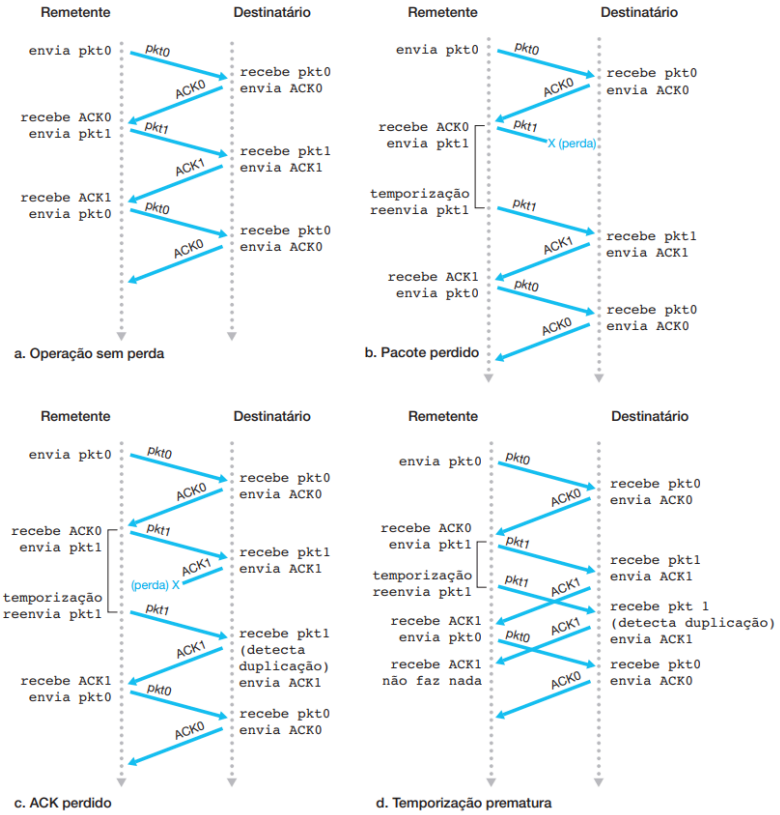
Transferência confiável de dados por um canal com erros de bits: rdt2.0

Agora preocuparemos, também, com a perda de pacote

Suponha que o remetente mandou uma mensagem e o que foi perdido é um dos pacotes ou o ACK do destinatário.

O remetente pode fazer uma escolha ponderada de tempo dentro do que seria provável que a perda tivesse ocorrido, e se não fosse recebido um ACK nesse tempo, retransmitiria o dado.

Pacotes duplicados por causa de um grande atraso poderiam ocorrer, mas o identificador do pacote ajuda o destinatário a cuidar disso.



Com um pacote de qualquer tamanho, o temporizador começa somente após a saída do último bit, e a resposta do destinatário só virá após ele receber o pacote por completo.

3.4.2 Protocolos de transferência confiável de dados com paralelismo

Suponha a transmissão de dados por meio da fibra óptica entre as costas dos EUA com RTT = 30 ms (15 ms para ir e 15 ms para voltar)

R = 1 Gbit/s e L = 8 mil bits tempo de transmissão = L/R = 8 micro segundos para que todos os dados saiam do computador de origem

15 ms para que o dado atravesse o país

Dessa forma, o último bit chega do outro lado com 15 ms + L/R = 15,008 ms

Verificamos que o remetente ficou mais de 99,999% do tempo só esperando uma resposta, pois um pacote só é enviado quando há a confirmação do anterior, por causa do pare e espere.

Para resolver esse problema, assim que um pacote acaba de ser enviado, um próximo já começa a ser, aproveitando toda a rede

3.4.3 Go-Back-N (GBN)

O remetente é autorizado a emitir múltiplos pacotes sem esperar reconhecimento

Suponha que ele vai enviar 20 pacotes, os números de cada um serão os do intervalo [0, total de pacotes - 1]. Porém há uma quantidade N de pacotes que podem ser enviados por vez.

Ao enviar um pacote, o próximo da fila a ser enviado é marcado como sendo o **nextseqnum**

Quando um pacote chega no destino, é retornado um ACK com um valor indicando o último pacote que foi reconhecido. Dessa forma, definiremos **base** como sendo o número do pacote mais velho que ainda não obteve resposta.

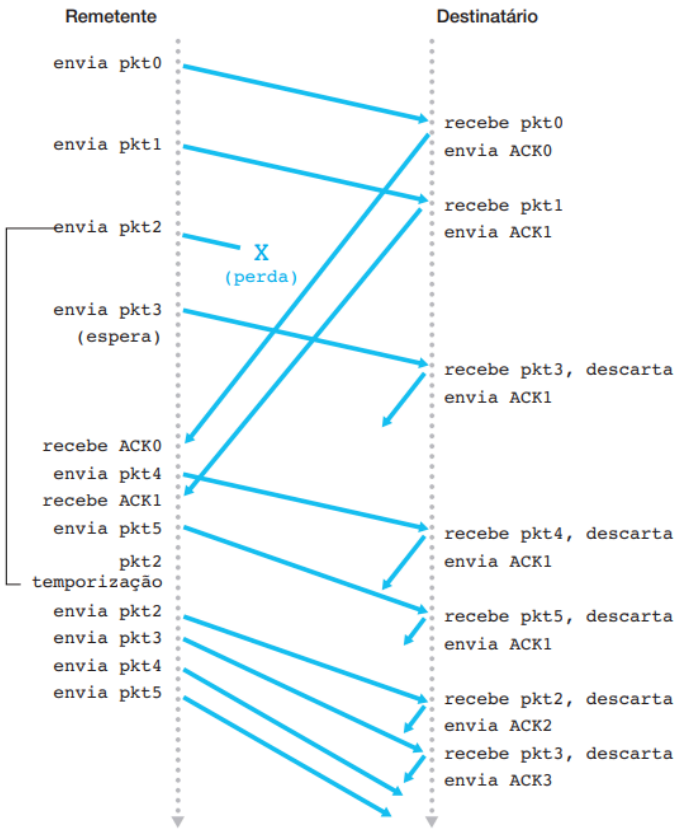
[0, base-1] são todos os pacotes enviados e reconhecidos. [base, nextseqnum-1] são todos os pacotes enviados que esperam para serem reconhecidos.

N pacotes podem ser enviados de uma só vez. Supondo que de 0 a 4 tudo corra bem, enviaremos 5 pacotes e receberemos 5 ACKs, cada um indicando que o pacote chegou. Se um ACK não chegou, contanto que o seguinte chegue, não há problemas, pois os pacotes são entregues de forma sequencial. Assim que o ACK do pacote mais antigo chega, a janela vai para o lado e permite o envio do nextseqnum.

Caso um pacote chegue ao destinatário fora de ordem, ele será descartado e um ACK com o mais recente que chegou corretamente será retornado, indicando até onde os dados foram recebidos corretamente, dessa forma a janela pode ser deslizada até o correto. Se de 0 a 3 veio em ordem, mas o 4 não, o primeiro da janela se tornará o 4.

Caso sejam enviados 5 pacotes e o primeiro seja perdido, todos os outros são descartados, pois estão fora de ordem

Há uma temporização esperando o ack do que possui um erro, e caso ele não chegue, como irá acontecer, todos os pacotes desde base até base+N-1 serão reenviados



Dependendo da situação, pode haver muitos pacotes pendentes, o que pode fazer o GBN ficar muito lento, pois muitos pacotes deverão ser retransmitidos sem necessidade.

3.4.4 Repetição seletiva (SR)

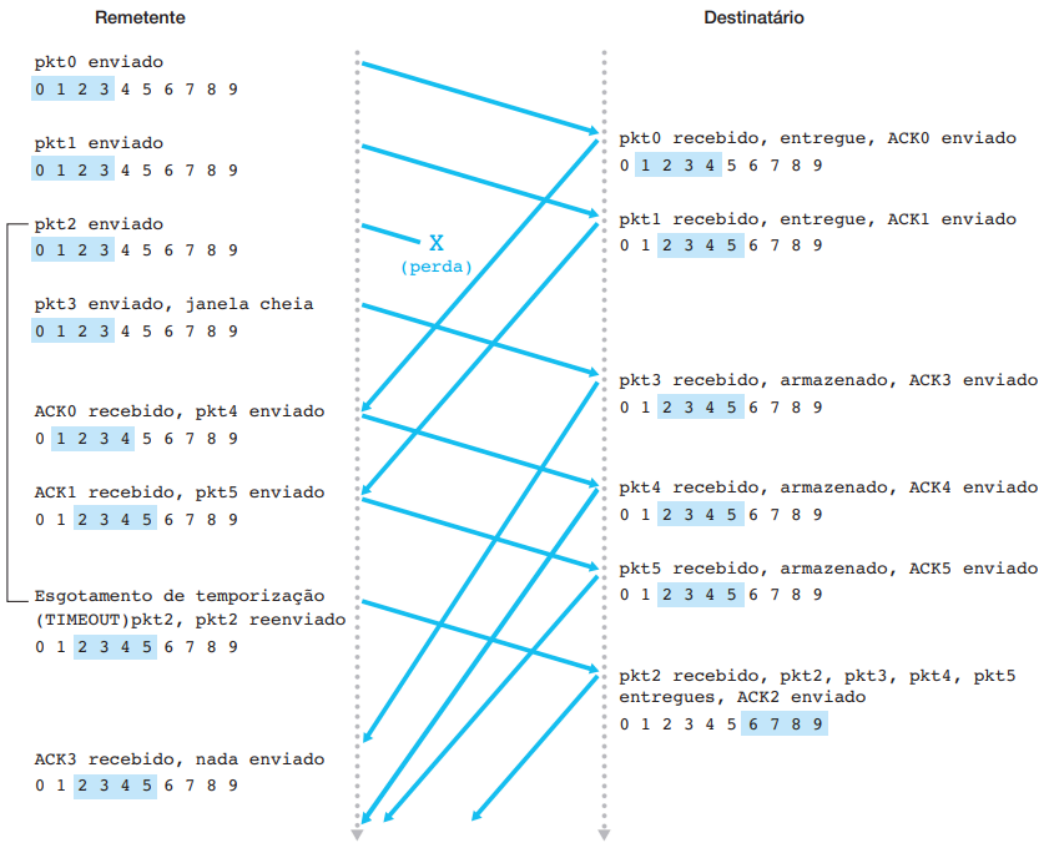
Somente os pacotes suspeitos de terem sido perdidos ou recebidos com erro serão retransmitidos.

Faz com que o destinatário reconheça individualmente os pacotes recebidos de modo correto

Uma janela de tamanho N limita o número de pacotes pendentes não reconhecidos dentro da rede.

O destinatário irá reconhecer um pacote recebido caso ele esteja ou não na ordem certa.

Os pacotes que são recebidos fora de ordem vão para um buffer esperar os atrasados. Assim que eles chegam, os dados são entregues para a camada de cima



A janela só se desloca quando o pacote mais antigo é entregue e confirmado. Cas contrário ela ficará parada, mesmo que todos os outros já foram confirmados

Como um pacote pode chegar atrasado, os ACKs não são cumulativos.

3.5 Transporte orientado para conexão: TCP

3.5.1 A conexão TCP

TCP é **orientado a para conexão,** pois os processos se apresentam enviando segmentos preliminares para iniciar variáveis de estado.

O estado das conexões reside nos sistemas finais.

A conexão TCP é um serviço full-duplex onde somente dois sistemas trocam mensagens.

Quem inicia a conexão é o cliente e o outro é o servidor.

O cliente envia um segmento especial, o servidor responde com outro e o cliente envia um terceiro, formalizando a conexão por uma **apresentação de três vias**.

Com a conexão ativa, o cliente passa uma cadeia de dados para o socket, que os envia para o buffer de envio, que são enviados de acordo com a conveniência do TCP.

A quantidade máxima que um segmento pode ter é limitada pelo **tamanho máximo do segmento** (MSS). O tamanho de um pacote é o MSS mais o tamanho do cabeçalho.

Um **segmento TCP** combina dados do cliente com um cabeçalho TCP, que é passado para a camada de rede para ser encapsulado e enviado.

Quando o TCP recebe um segmento, ele é colocado no buffer de recepção da conexão, que é lido pela aplicação

3.5.2 Estrutura do segmento TCP

O campo de dados contém dados da aplicação, sendo que os dados podem ser fragmentados para terem o tamanho do MSS.

Campos do TCP

- número de sequência e número de reconhecimento: usados pelo remetente e destinatário para uma conexão confiável
- janela de recepção: Controla o fluxo
- comprimento do cabeçalho: Pode ter tamanho variado por causa do campo de opções
- campo de opções: Possui tamanho variado e é usado quando é negociado o MSS
- flag: bit ack e outros

Número de sequência e números de reconhecimento

Quando uma mensagem é enviada e ela é muito grande, ela é dividida em segmentos e o **número de sequência** é o número do primeiro byte de cada segmento.

Cada segmento que chega do hospedeiro B tem um número de sequência para os dados que fluem de B para A

O **número de reconhecimento** que A atribui ao seu segmento é o número de sequência do próximo byte que ele está aguardando do B.

Se A recebeu os bytes de 0 a 535 e de 900 a 1000, a resposta que ele dará a B no campo de número de reconhecimento é o 536. Dessa forma, o TCP só reconhece os bytes até o primeiro que está faltando, provendo reconhecimentos cumulativos.

Neste exemplo, os dados chegaram fora de ordem, então eles são armazenados em um buffer.

3.5.3 Estimativa do tempo de viagem de ida e volta e de esgotamento de temporização

Estimativa do tempo de viagem de ida e volta (RTT)

RTT é o tempo entre o momento que o segmento é enviado e o momento em que é recebido um reconhecimento para ele.

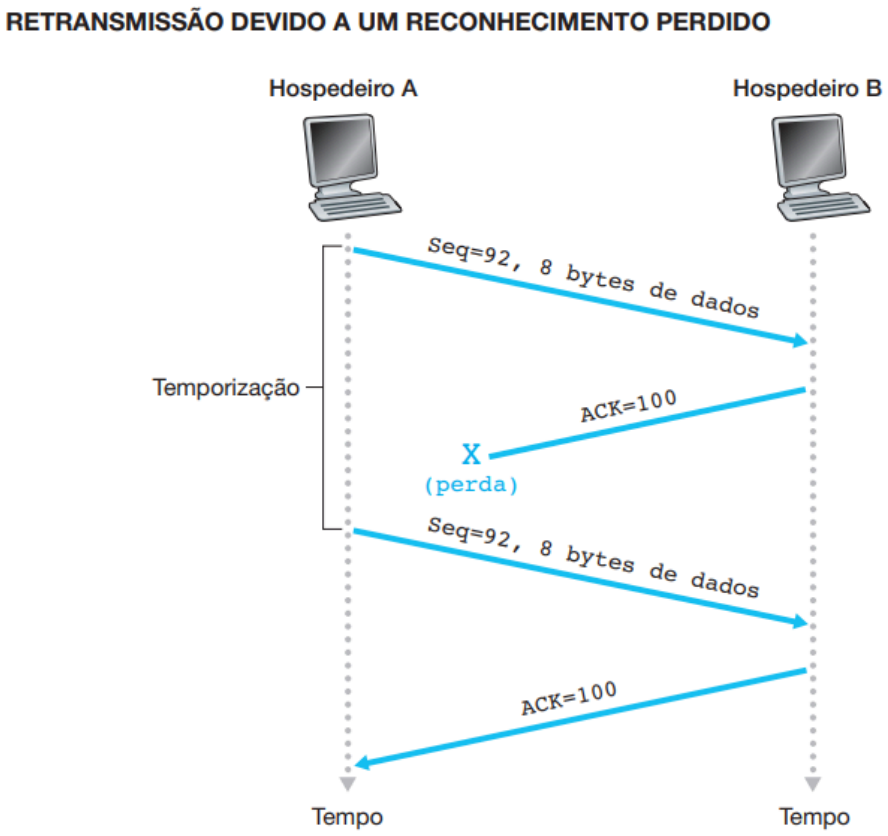
O RTT é estimado várias vezes durante a transmissão de um segmento, sendo calculado a partir do último valor, gerando uma média móvel exponencial ponderada.

O tempo utilizado para a temporização de retransmissão do TCP é maior ou igual ao RTT estimado calculado

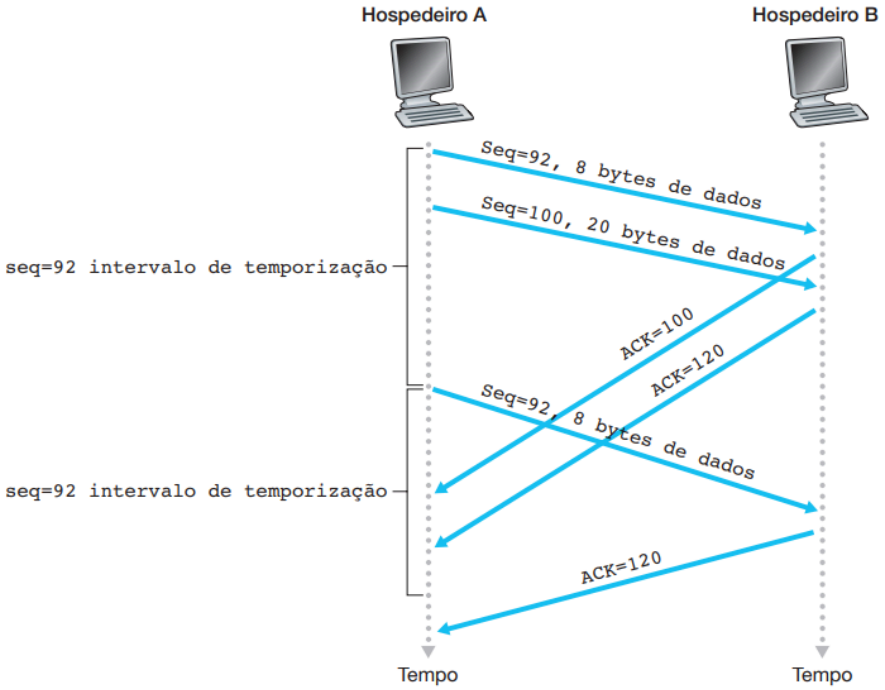
3.5.4 Transferência confiável de dados

Principais eventos relacionados com a transmissão e retransmissão de dados TCP:

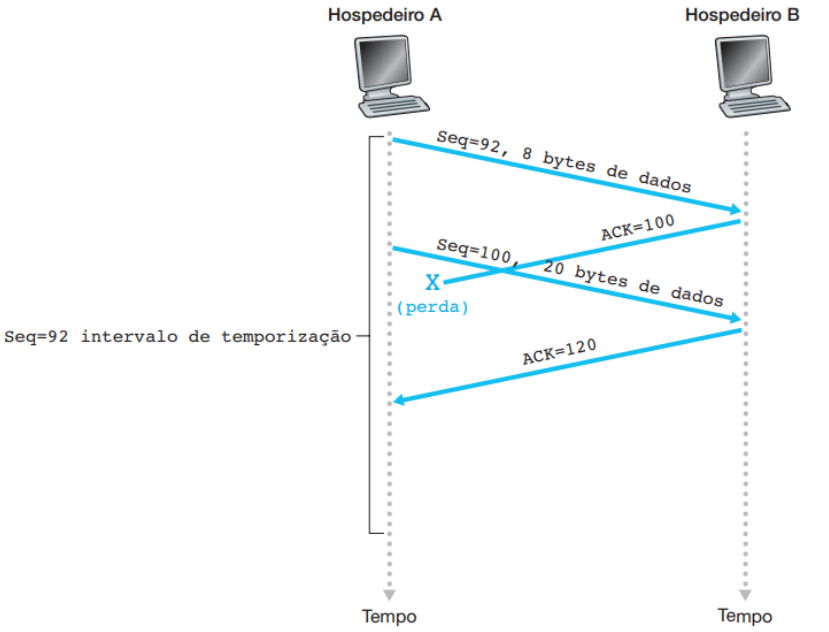
- Dados recebidos da aplicação: Os dados chegam da camada de aplicação, passam pelo TCP, são encapsulados e enviados ao IP
- esgotamento do temporizador: Retransmite o dado que causou o esgotamento e reinicia o temporizador
- recebimento do ACK: Recebe um ACK, verifica se é de um pacote a frente do que já foi enviado e reconhece os outros pacotes



SEGMENTO 100 NÃO RETRANSMITIDO



UM RECONHECIMENTO CUMULATIVO EVITA RETRANSMISSÃO DO PRIMEIRO SEGMENTO



Retransmissão rápida

ACK duplicado é um ACK que o remetente recebeu, mas que já foi anteriormente recebido, dessa forma, sabemos que houve perda de pacote.

Caso seja recebido um ACK repetido três vezes, o remetente não irá esperar acontecer o timeout do temporizador para reenviar os dados, pois assim que ele perceber, já irá reenviar o pacote que não foi reconhecido.

O TCP utiliza um **reconhecimento seletivo**, permitindo o destinatário reconhecer seletivamente segmentos fora de ordem, porém o ACK retornado é do último pacote que foi corretamente identificado. Dessa forma, há uma mescla entre o GBN e SR.

3.5.5 Controle de fluxo

Evita que o remetente estoure o buffer do destinatário ao compatibilizar a taxa na qual o remetente enviará os dados com a qual o destinatário consegue ler.

O remetente possui uma variável denominada de **janela de recepção**, usada para ter uma noção do espaço de buffer livre no destinatário.

O tamanho da janela de recepção (rwnd) é dada por RecvBuffer - [LastByteRcvd - LastByteRead]

O valor de rwnd é dinâmico, e pode ficar em zero, dessa forma, o destinatário não pode receber nada, mas como ele não retorna nada, então o remetente deve continuar mandando informações esperando o buffer esvaziar.

3.5.6 Gerenciamento da conexão TCP

Etapas para a criação de uma conexão TCP de três vias

1. O cliente envia um TCP ao servidor somente com as flags do cabeçalho e o bit SYN em 1
2. O servidor recebe, extrai os dados e aloca buffers e variáveis TCP e envia uma confirmação de aceitação ao cliente. SYN continua em 1
3. O cliente recebe a resposta e reserva buffers e variáveis para a conexão e envia mais um segmento ao servidor, confirmando, de fato, a conexão, mas agora com o SYN em 0

Para encerrar uma conexão, o cliente envia uma mensagem com o FIN em 1, o servidor reconhece e retorna um ACK. Depois o servidor faz o mesmo. Dessa forma, a conexão é encerrada e os recursos são liberados.

3.5 Princípios de controle de congestionamento

3.6.1 As causas e os custos do congestionamento

3.6.2 Mecanismos de controle de congestionamento

Controle de congestionamento fim a fim: A camada de rede não oferece suporte explícito à camada de transporte. O TCP deve percebe por meio de perdas de pacotes e atrasos que a rede está congestionada.

Controle de congestionamento assistido pela rede: Os roteadores fornecem informações ao remetente que a rede está congestionada, podendo ser a partir de um único bit

3.7 Controle de congestionamento no TCP

O TCP usa controle de congestionamento de fim a fim.

Se um remetente percebe que há pouco congestionamento entre ele e o destinatário, aumenta a taxa de envio, caso contrário, diminui.

O mecanismo de controle de congestionamento que está no remetente monitora uma variável de **janela de congestionamento**, chamada de cwnd, e limita a taxa na qual o TCP envia tráfego para dentro da rede

A taxa de envio de dados pelo remetente é aproximadamente cwnd/RTT bytes por segundo

Um **evento de perda** em um remetente TCP ocorre quando há o timeout ou o recebimento de três ACKS repetidos

O TCP é autorregulado por aumentar rapidamente a janela quando há confirmações rápidas e aumentar devagar quando elas demoram, seguindo as seguintes propriedades:

- Quando há um um evento de perda, a taxa de envio deve diminuir
- Um segmento reconhecido indica que a rede está enviando corretamente e a janela pode aumentar
- A taxa vai sendo aumentada até que ocorra alguma perda, depois ela abaixa

O **algoritmo de controle de congestionamento TCP** possui três componentes principais: 1) Partida lenta, 2) Contenção de congestionamento e 3) Recuperação rápida.

Partida lenta

O valor de cwnd começa muito baixo, cerca de 1 MSS, e a cada RTT , o tamanho de cwnd é duplicado a cada rodada de envio e na rodada em que ele ultrapassa o sshtresh, o seu valor é setado como o próprio sshtresh

SSHTRESH é a metade do valor da última taxa com um envio correto

Prevenção de congestionamento

Quando cwnd atinge a metade do valor que antes ocorreu um erro, ou seja, atinge o sshtresh, então o crescimento passa a ser linear., ou seja, de uma em uma unidade

Recuperação rápida

TCP Tahoe: Corta pela metade o sshthresh quando identifica uma duplicação tripla do ACK. A janela sempre irá para o tamanho 1

TCP Reno: Quando ocorre 3 ACK's duplicados, a janela vai para a metade +3. Quando ocorre timeout a janela vai para 1.

Descrição macroscópica da vazão do TCP

Quando o tamanho de uma janela for w bytes, a taxa de transmissão será w/RTT

Capítulo 4 - A camada de rede

4.1 Introdução

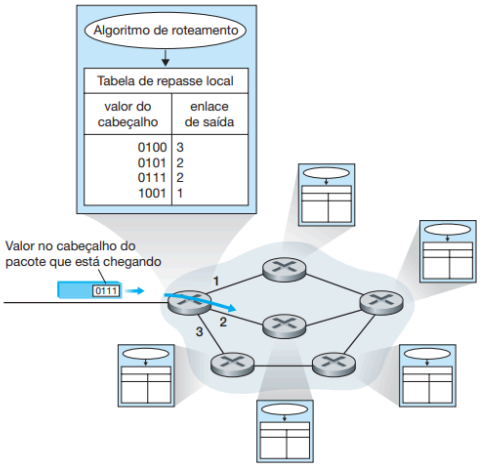
4.1.1 Repasse e roteamento

A camada de rede transporta pacotes de um remetente a um destinatário, executando:

- Repasse: Quando um pacote chega a um roteador, este deve levá-lo à saída correta
- Roteamento: Determina a rota ou caminho tomado pelos pacotes

Cada roteador possui uma **tabela de repasse**, que analisa um valor no cabeçalho que está chegando e determina para qual saída o pacote deve ser encaminhado

FIGURA 4.2 ALGORITMOS DE ROTEAMENTO DETERMINAM VALORES EM TABELAS DE REPASSE



4.1.2 Modelos de serviço de rede

Fornece um modelo de serviço de **melhor esforço**, pois simplesmente recebe o pacote e o repassa, sem nenhuma garantia.

4.2 Redes de circuitos virtuais e de datagramas

Redes de circuitos virtuais (Redes CV): Redes que oferecem apenas um serviço orientado para conexão na camada de rede

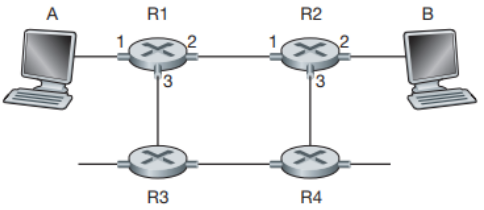
Redes de datagramas: Redes que oferecem apenas um serviço não orientado para conexão na camada de rede

4.2.1 Redes de circuitos virtuais

Um circuito virtual consiste em

1. Enlaces e roteadores entre origem e destino
2. Número de CV para cada enlace ao longo do caminho
3. Registros na tabela de repasse

FIGURA 4.3 UMA REDE DE CIRCUITOS VIRTUAIS SIMPLES



Um pacote vai passar pelo caminho A-R1-R2-B. Sai de A com CV 12, depois muda para 22 e no R2 vira 32.

O número de CV muda a cada enlace a partir da tabela de repasse, que terá um item adicionado

Os roteadores armazenam **informação de estado de conexão** para as conexões em curso. Se uma conexão é ativada, um registro é adicionado, se for desativada, é removido.

Fases de um CV:

- Estabelecimento do CV: A camada de rede determina por onde todos os pacotes passarão, o número CV para cada enlace e adiciona um registro em cada tabela de repasse
- Transferência de dados
- Encerramento do CV: Remetente ou destinatário avisam que desejam desativar o CV. É enviado um aviso até o outro lado e os registros são removidos das tabelas

As **mensagens de sinalização** são as responsável por avisar aos roteadores que há um CV ou um deixou de existir

4.2.2 Redes de datagramas

Quando um host quer enviar um pacote, ele marca o endereço final e envia para a rede.

Cada roteador usa o endereço destino para repassar os dados, onde cada um tem uma tabela de repasse para mapear os endereços.

Faixa de endereços de destino	Interface de enla
11001000 00010111 00010000 00000000 até 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 até 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 até 11001000 00010111 00011111 11111111	2
senão	3

Prefixo do endereço	Interface de enlace
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
senão	3

O roteador compara um prefixo do endereço do destino do pacote com os registros na tabela e transmite para aquele que houver uma concordância

Quando há várias concordâncias, o roteador usa a **regra de concordância do prefixo mais longo**, ou seja, o prefixo com mais bits iguais.

De minutos em minutos a tabela de repasse é atualizada, dessa forma, alguns pacotes podem passar por outros caminhos e acabarem chegando fora de ordem.

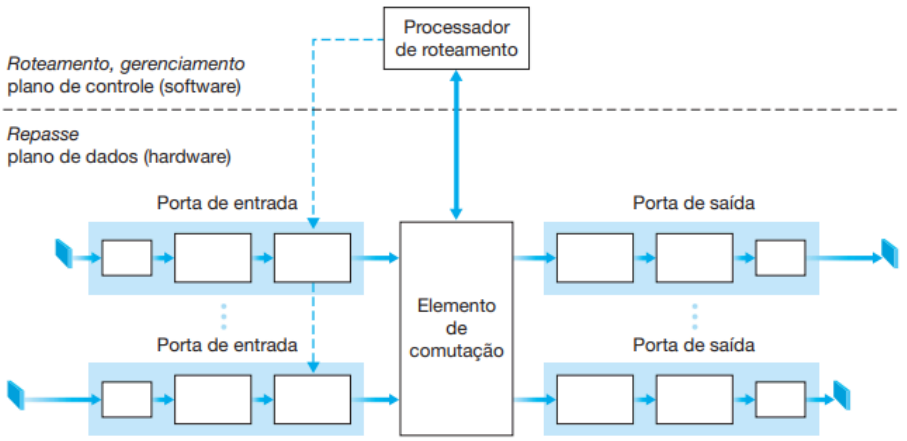
Diferenças entre redes de circuitos virtuais e redes de datagramas

As CVs iniciam uma conexão reservando recursos e determinam por onde todos os pacotes deverão passar até chegar no destinatário, garantindo a sua chegada.

A rede de datagrama decide por onde cada pacote vai passar de acordo com informações do tráfego durante o percurso dos dados

4.3 O que há dentro de um roteador?

FIGURA 4.6 ARQUITETURA DE ROTEADOR



- Portas de entrada: Conecta um enlace ao roteador, verifica a tabela de repasse e determina a porta de saída
- Elemento de comutação: Conecta a porta de entrada com a de saída (Barramento)
- Portas de saída: É por onde os dados irão sair
- Processador de roteamento: Executa protocolos de roteamento, mantém as informações da tabela de repasse e faz o gerenciamento da rede.

Plano de repasse do roteador: Funções de repasse que são executadas via hardware do processador de roteamento

4.3.1 Processamento de entrada

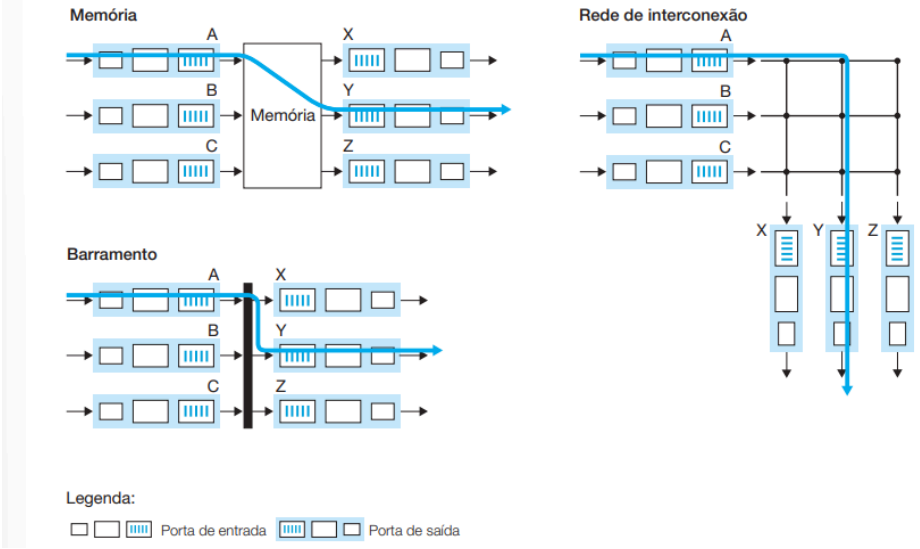
Utiliza a tabela de repasse para saber para qual porta de saída os dados irão.

Cada porta de entrada possui uma cópia da tabela de repasse, calculada e atualizada pelo processador de roteamento

Quando uma porta de saída é definida a partir de uma pesquisa na tabela, o pacote vai para o elemento de comutação, mas caso ele esteja ocupado, começa a enfileirar os pacotes

4.3.2 Elemento de comutação

FIGURA 4.8 TRÊS TÉCNICAS DE COMUTAÇÃO



Comutação por memória

O pacote entra no roteador, gera uma interrupção, é processado extraindo o endereço de destino do cabeçalho, consulta a porta de saída apropriada e copia o pacote para os buffers na porta de saída.

Só pode ser repassado um pacote por vez devido a escrita e leitura na memória

Comutação por barramento

Transfere os pacotes diretamente da porta de entrada para a porta de saída. É enviado um rótulo junto ao pacote. Todas as portas recebem o mesmo pacote, mas só a que combina mantém o pacote. Depois remove o rótulo e envia o pacote

Se N pacotes chegarem em diferentes portas de entradas, N-1 deverão esperar para que um deles passe pelo barramento.

Comutação por uma rede de interconexão

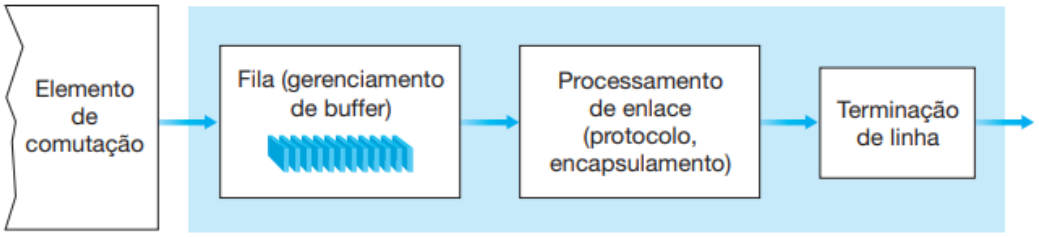
Possui 2N barramentos. N barramentos de entrada(horizontal) mais N barramentos de saída (vertical). Os cruzamentos podem ser abertos ou fechados, de acordo com o que o elemento de comutação mandar.

Se um pacote vem da porta A para Y, o seu cruzamento é fechado. Assim, pacotes com diferentes portas de entrada e saída podem se comunicar ao mesmo tempo. Por exemplo: A para Y e B para X

Caso um pacote vá para a mesma porta de saída, ele deverá esperar.

4.3.3 Processamento de saída

FIGURA 4.9 PROCESSAMENTO DE PORTA DE SAÍDA



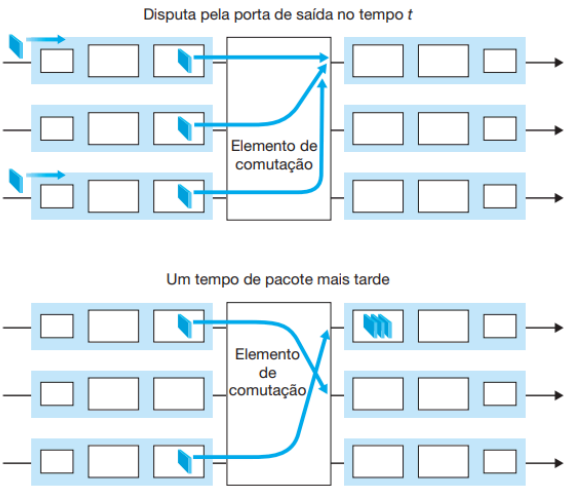
4.3.4 Onde ocorre a formação da fila?

As filas podem ser formadas tanto na porta de entrada como na porta de saída

Quando mais elementos chegam além da capacidade, são geradas filas para lidar com os dados. As filas na entrada são para vários itens usarem a porta para entrar e a de saída é quando vários necessitam de sair pelo mesmo local.

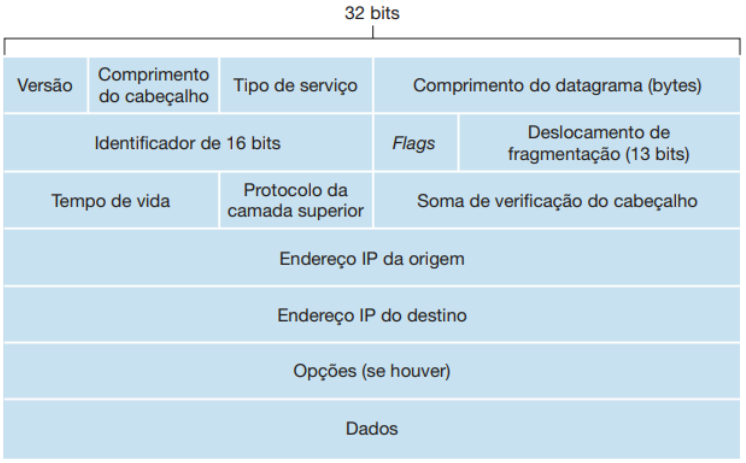
Um escalonador é usado para escolher qual pacote irá sair, podendo usar um tipo de FIFO ou uma fila ponderada justa, que compartilha o enlace de saída com equidade entre as diferentes conexões.

FIGURA 4.10 FORMAÇÃO DE FILA NA PORTA DE SAÍDA



4.4 O protocolo da internet (IP): Repasse e endereçamento na internet

4.4.1 Formato de datagrama



Fragmentação do datagrama IP

Diferentes roteadores possuem diferentes capacidades (MTU) para lidar com grandes quantidades de dados, gerando incongruência nos tamanhos, além de que diferentes protocolos podem ser usados.

Quando um roteador A identifica que o B não comporta o tamanho, ele divide o datagrama em fragmentos, sendo que no sistema final o fragmento é remontado na camada de rede, assim o TCP e UDP ficam intactos.

Quando o datagrama é criado, ele é marcado com um ***número de identificação***. Se ele é quebrado em fragmentos, o valor da flag vai para 1 e o último pacote fica com a ***flag*** em zero. Assim, o sistema final identifica uma repetição nos endereços e no identificador e consegue remontar o pacote a partir do ***campo de deslocamento.***

TABELA 4.2 FRAGMENTOS IP

Fragmento	Bytes	ID	Deslocamento	*Flag*
1º fragmento	1.480 bytes no campo de dados do datagrama IP	identificação = 777	0 (o que significa que os dados devem ser inseridos a partir do byte 0)	1 (o que significa que há mais)
2º fragmento	1.480 bytes de dados	identificação = 777	185 (o que significa que os dados devem ser inseridos a partir do byte 1.480. Note que 185 x 8 = 1.480)	1 (o que significa que há mais)
3º fragmento	1.020 bytes de dados (= 3.980 –1.480 –1.480)	identificação = 777	370 (o que significa que os dados devem ser inseridos a partir do byte 2.960. Note que 370 x 8 = 2.960)	0 (o que significa que esse é o último fragmento)

4.4.2 Endereçamento IPv4

Uma interface é uma conexão entre o hospedeiro e o meio físico. O roteador tem diferentes interfaces, uma para cada enlace.

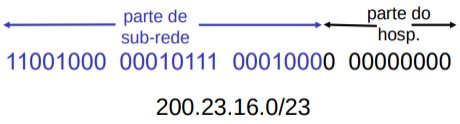
Um endereço IP está associado a uma interface e não a um hospedeiro ou roteador que possui a interface.

Cada IP tem 32 bits, ou 4 bytes

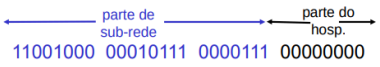
Cada interface deve ter um IP exclusivo e não pode ser escolhido de qualquer maneira, sendo que uma parte do mesmo é determinada pela rede que está conectada.

Uma sub-rede é uma rede que interconecta diferentes interfaces dentro da mesma rede.

a.b.c.d/Y : Os Y bits à esquerda indicam o endereço da sub-rede, sendo Y seu tamanho



200.23.16.0/23

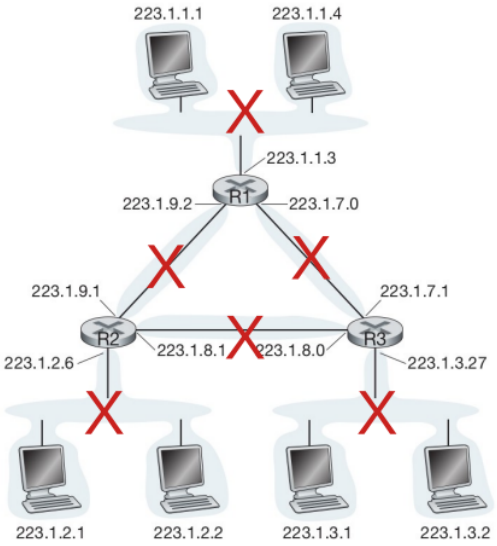


200.23.7.0/24

- 200.23.7.0 define o endereço da rede e não é utiizado por nenhuma interface
- 200.23.7.255 define o endereço de broadcast dentro da subrede e também não é utilizado por nenhuma interface

- Redes privadas
  - 10.0.0.0 até 10.255.255.255
  - 172.16.0.0 até 172.31.255.255
  - 192.168.0.0 até 192.168.255.255
  -
- Esses endereços não funcionam na Internet
  - Funcionam apenas em redes locais

Para determinar a quantidade de sub-redes, devemos isolar cada interface



O endereço de broadcast 255.255.255.255 envia um datagrama para todos os hospedeiros em uma sub-rede.

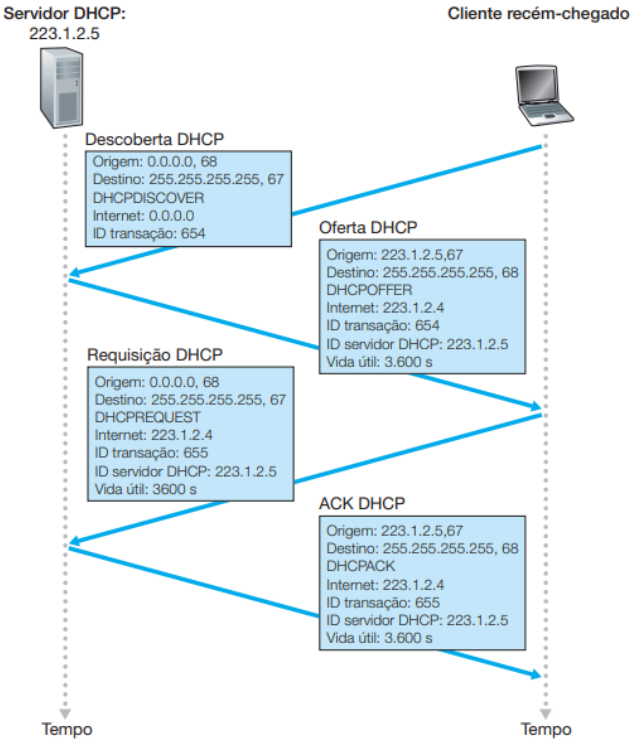
Obtenção de um endereço de hospedeiro: Protocolo de configuração dinâmica de hospedeiros (DHCP)

Permite um hospedeiro obter um IP de forma automática. Podendo receber um IP igual ou temporário toda vez que se conectar.

O DHCP possui 4 etapas, sendo elas:

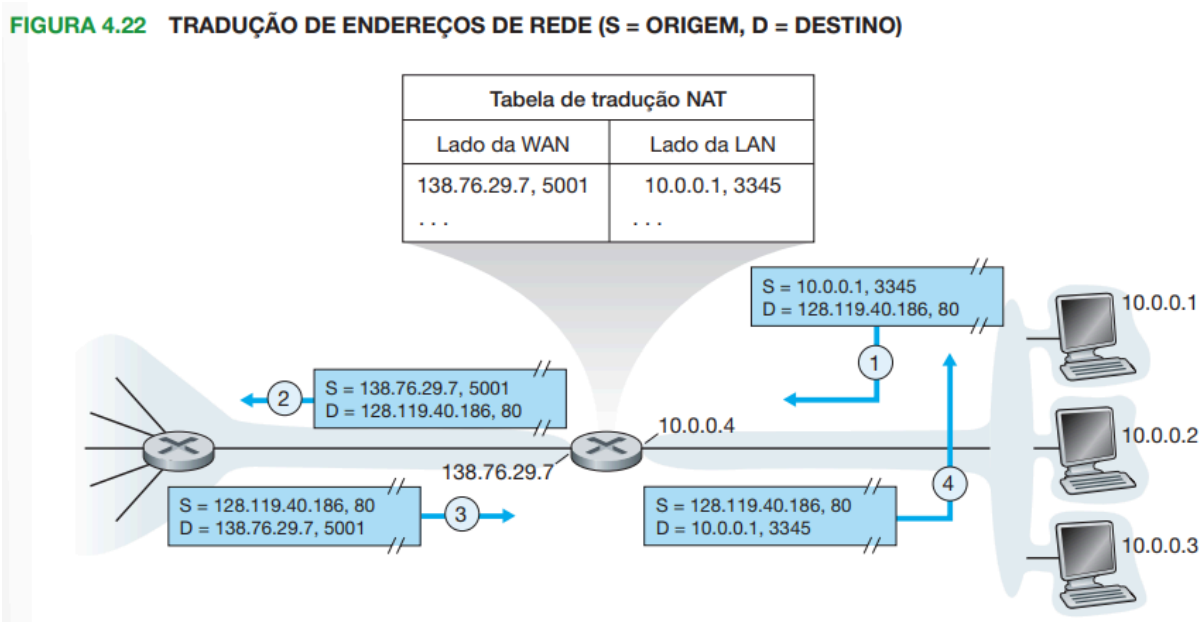
1. Descobrir o servidor DHCP: Envia um UDP para a porta 67 para o IP 255.255.255.255 e como destinatário o 0.0.0.0.
2. Ofertas dos servidores DHCP: O servidor recebe uma mensagem de descoberta e envia um oferta de IP para o 255.255.255.255. E cada mensagem possui seu ID, endereço proposto, máscara e tempo de concessão do IP
3. Solicitação DHCP: O cliente envia uma mensagem de solicitação aceitando uma das ofertas
4. DHCP ACK: O servidor confirma enviando um DHCP ACK e recebe, também, o IP do servidor DNS, IP do gateway padrão e máscara de sub-rede

FIGURA 4.21 INTERAÇÃO CLIENTE-SERVIDOR DHCP



Tradução de endereços na rede (NAT)

FIGURA 4.22 TRADUÇÃO DE ENDEREÇOS DE REDE (S = ORIGEM, D = DESTINO)



O roteador que usa NAT possui um único endereço IP

O que sai da rede residencial (direita) tem o endereço de origem 138.76.29.7, e todos os dados que chegam nesta rede terão como destino esse mesmo endereço

A tabela de tradução NAT contém nos registros da tabela o número das portas e o endereço IP.

- O host (10.0.0.1) escolhe uma porta e envia um datagrama para dentro da LAN.
- O roteador recebe, e altera o endereço de origem para o seu e adiciona uma nova porta, então repassa os dados.
- O servidor processa os dados e envia para o NAT.
- O NAT usa os dados que possui e identifica o destinatário correto.

A NAT impede conexões P2P, pois caso um dos lados esteja sob uma NAT, não poderia aceitar conexões TCP.

Com o uso de um intermediário, com a conexão estabelecida, então podemos ter depois a conexão direta entre os pares

UPnP (Universal Plug and Play)

Permite o hospedeiro a descobrir e configurar uma nat próxima, permitindo conexões TCP e UDP entre hospedeiros

A aplicação solicita ao NAR um mapeamento entre seu endereço privado e público. Se a NAT aceitar, cria uma conexão e nós externos podem se comunicar, pois o UPnP deixa a aplicação conhecer o IP e número de porta pública.

4.4.3 Protocolo de Mensagens de Controle da Internet (ICMP)

Usado entre hospedeiros e roteadores para comunicar informações da camada de rede entre si, principalmente para erros.

Se em algum ponto um roteador IP não conseguiu descobrir o caminho para o hospedeiro. Então o roteador cria e envia uma mensagem ICMP do tipo 3 ao seu host, indicando o erro.

Está acima do IP, pois suas mensagens são carregadas dentro de datagramas IP.

Quando um datagrama IP é recebido com uma mensagem ICMP, ela possui um campo de tipo e código, além do IP que causou o erro.

FIGURA 4.23 TIPOS DE MENSAGENS ICMP

Tipo ICMP	Código	Descrição
0	0	resposta de eco (para *ping*)
3	0	rede de destino inalcançável
3	1	hospedeiro de destino inalcançável
3	2	protocolo de destino inalcançável
3	3	porta de destino inalcançável
3	6	rede de destino desconhecida
3	7	hospedeiro de destino desconhecido
4	0	repressão da origem (controle de congestionamento)
8	0	solicitação de eco
9	0	anúncio do roteador
10	0	descoberta do roteador
11	0	TTL expirado
12	0	cabeçalho IP inválido

4.5 Algoritmos de roteamento

Um hospedeiro está ligado diretamente a um **roteador default**, que é o primeiro a receber um pacote emitido

Em um grafo, os nós são roteadores e as arestas são os enlaces.

Algoritmo de roteamento global: Calcula o menor caminho usando conhecimento completo da rede, podendo ser realizado em um local ou replicado para vários

Algoritmo de roteamento descentralizado: Cálculo realizado de modo iterativo e distribuído. Cada nó só sabe o custo do seu vértice adjacente e a partir da troca de informação com seus vizinhos descobre o custo para outros

Algoritmos estatísticos: A rota muda muito raramente, em geral, com interferência humana

Algoritmos dinâmicos: Muda as rotas junto com a modificação da rede ou tráfego

4.5.1 O algoritmo de roteamento de estado enlace (LS)

A topologia e todos os custos do enlace são conhecidos e é um algoritmo global

Cada nó transmite informações a todos os outros contendo informações sobre o enlace

O algoritmo usado é o de Dijkstra

4.5.2 O algoritmo de roteamento de vetor de distâncias (DV)

É iterativo, assíncrono e distribuído.

Cada nó recebe informação dos seus vizinhos e distribui seus resultados, sendo realizado até que mais nenhuma troca seja realizada, e não requer que seja executado ao mesmo tempo para todos.

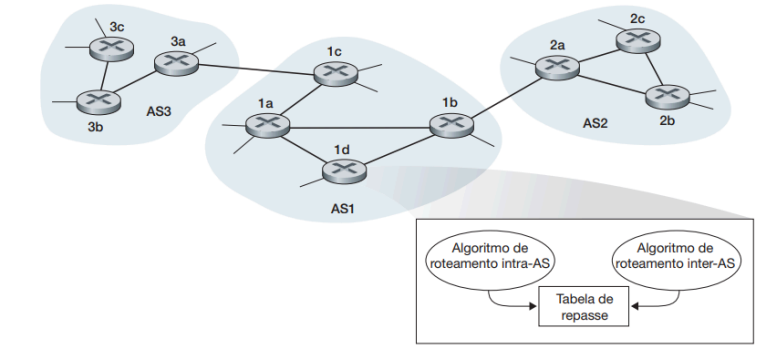
Utiliza o algoritmo de Bellman-Ford

4.5.3 Roteamento hierárquico

Os roteadores não utilizam todos os mesmos algoritmos para atualizar as tabelas de repasse

Os roteadores são agrupados em sistemas autônomos, operados pelos ISPs. Assim cada grupo executa um algoritmo, sendo denominado de protocolo de roteamento intrasistema autônomo. Cada sistema autônomo é conectado por roteadores denominados de roteadores de borda (gateway routers)

FIGURA 4.32 UM EXEMPLO DE SISTEMAS AUTÔNOMOS INTERCONECTADOS



Quando um AS (Sistema autônomo) recebe um pacote, ele sabe o melhor caminho para os roteadores internos e para o gateway, dessa forma o envia e cabe ao próximo AS lidar com o pacote.

O protocolo de roteamento inter-AS rodam o protocolo BGP4, que permite um roteador interno saber para onde cada roteador de borda irá levar os dados, permitindo, assim, o envio dos dados de forma correta em caso de mais de um gateway.

4.6.1 Roteamento intra-AS na internet: RIP

Disponível para ISPs de níveis mais baixos e para redes corporativas

Protocolo de roteamento intra-AS determina como o roteamento é rodado dentro de um AS. Também são conhecidos como IGP (interior gateway protocol)

O RIP foi um dos primeiros protocolos de roteamento intra-AS

Funciona de forma similar ao protocolo DV (Bellman-Ford), sendo que usa a contagem de saltos (sub redes percorridas) como custo, onde cada enlace tem custo 1, sendo definidos desde a origem até o roteador destino

O custo máximo para um caminho é 15. As atualizações de caminhos ocorrem com cerca de 30 s entre vizinhos, com uma mensagem contendo até 25 sub-redes de destino dentro da AS, junto com suas distâncias entre elas.

Cada roteador tem uma tabela RIP chamada de tabela de roteamento, que inclui o vetor de distâncias e a tabela de repasse desse roteador. A tabela possui uma coluna para a sub-rede destino, o indicador do próximo roteador e a quantidade de saltos até o destino.

Se um roteador não ouvir nada de seu vizinho em 180 s, isso indica que o mesmo está fora do ar. A tabela é modificada e a informação é propagada para os vizinhos

Executa sobre UDP

4.6.2 Roteamento intra-AS na internet: OSPF

Está disponível em ISPs de níveis mais altos

O roteador monta um grafo de todo o AS e usa o Dijkstra para determinar o caminho para todas as sub-redes.

O administrador da rede pode escolher os custos dos enlaces.

O roteador manda seus dados para todos os roteadores no sistema autônomo. A cada 30 minutos transmite o estado do enlace ou quando ocorre alguma mudança. Sendo as mensagens transmitidas pelo IP com código de protocolo 89. para OSPF.

Possui uma troca de mensagens confiável, além de verificar se os enlaces estão operacionais com mensagens HELLO e permite um roteador ter acesso ao banco de dados do seu vizinho.

Características:

- Segurança: troca de mensagens autenticadas
- Caminhos múltiplos com mesmo custo: Evita que somente um caminho carregue todo o tráfego. o RIP só comporta um.
- Suporte integrado para roteamento individual (unicast) e em grupo (multicast)
- Hierarquia dentro de um único domínio de roteamento: Divide-se em áreas onde cada uma roda o seu próprio algoritmo de roteamento e um roteador transmite seus dados para os outros da área. Sendo que um ou mais são gateways de área. Uma área é a backbone (controla o tráfego entre as áreas)

4.6.3 Roteamento inter-AS: BGP

BGP 9 Border gateway protocol) é o padrão para roteamento entre sistemas autônomos na Internet, sendo que oferece a cada AS meios de:

- Obter informação de alcançabilidade de ASs vizinhos
- Propagar informação de alcançabilidade a todos os roteadores internos ao AS
- Determinar boas rotas para sub-redes

Permite que uma sub-rede seja conhecida pela internet

Há pares de roteadores (pares BGP) que trocam informações de roteamento por conexões TCP semipermanentes (sessão BGP)

A Sessão BGP interna é para roteadores dentro do mesmo AS e a externa é para diferentes AS.

Permite que cada AS conheça os caminhos alcançáveis por meio do seu AS vizinho. Sendo que os vizinhos não são hospedeiros, mas IPs com máscaras, para saber o seu prefixo.

Mensagens BGP

- Mensagens BGP trocadas usando TCP.
- Mensagens BGP:
  - OPEN: abre conexão TCP com par e autentica remetente
  - UPDATE: anuncia novo caminho (ou retira antigo)
  - KEEPALIVE: mantém conexão viva na ausência de UPDATES; também envia ACK para solicitação OPEN
  - NOTIFICATION: informa erros na msg anterior; também usada para fechar conexão

Atributos de caminho e rotas BGP

Um AS é identificado pelo seu número de sistema autônomo (ASN)

Quando um roteador anuncia um prefixo para uma sessão BGP, ele inclui vários atributos juntos

Seleção de rota do BGP

O BGP usa o eBGP e o iBGP para distribuir as rotas aos roteadores. Se houver duas ou mais rotas para o mesmo prefixo, o BGP irá realizar eliminações pelas seguintes regras:

- Selecionadas rotas que têm valores de preferência mais alta
- Selecionada a que possui o menor AS-PATH (Caminho que contém os ASs pelos quais passou o anúncio do prefixo)
- Selecionada o NEXT-HOP mais próximo
- Se ainda houver empate, usa identificadores BGP

Capítulo 5 - Camada de enlace física

5.1 Introdução à camada de enlace

Para que um datagrama seja levado de uma origem a um destino, ele deve passar por diferentes enlaces individuais existentes no caminho fim a fim.

5.1.1 Os serviços fornecidos pela camada de enlace

Enquadramento de dados: Encapsula cada datagrama da camada de rede (IP) dentro da camada de enlace antes de transmitir.

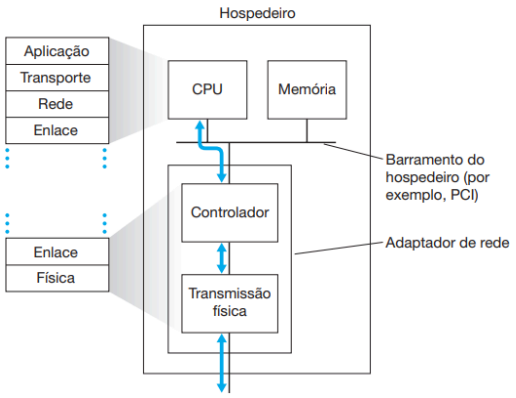
Acesso ao enlace: O MAC dita como os dados são enviados, coordenando transmissões múltiplas de muitos nós. O que não é necessário em uma conexão ponto a ponto

Entrega confiável: Garante o transporte do datagrama pelo enlace, muito usado em redes sem fio e pouco usado para redes de fibra.

Detecção e correção de erros: Detecta um erro para que o pacote não seja passado de forma desnecessária. O nó transmissor envia bits de detecção e o receptor verifica. O hardware pode ainda realizar a correção do erro sem necessitar de retransmissão.

5.1.2 Onde a camada de enlace é implementada?

FIGURA 5.2 ADAPTADOR DE REDE: SEU RELACIONAMENTO COM O RESTO DOS COMPONENTES DO HOSPEDEIRO E A FUNCIONALIDADE DA PILHA DE PROTOCOLOS



A camada de enlace é implementada em um adaptador de rede [placa de interface de rede (NIC)], possuindo em seu núcleo um chip para executar os serviços da camada.

Os hardwares são os que implementam protocolos como o Ethernet e o WIFI 802.11.

No transmissor, o controlador separa o datagrama criado, salva na memória, encapsula o datagrama e transmite para o enlace de comunicação, seguindo o seu protocolo. O receptor recebe o quadro e extrai o datagrama da rede, verifica erros e repassa o pacote.

É uma combinação de software e hardware, pois ao receber um datagrama, o software da camada de enlace verifica interrupções para lidar com os dados, lidos através de barramentos.

5.2 Técnicas de detecção e correção de erros

No nó remetente, os dados possuem bits de detecção e correção (error detection-and-correction EDC) adicionados.

Além do datagrama, endereçamento da camada de enlace, números de sequência e outros campos são protegidos.

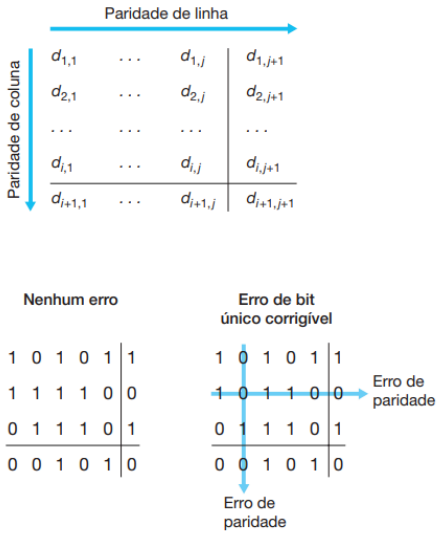
O receptor tenta verificar se os dados recebidos são iguais aos enviados, mas nem sempre é possível identificar erros, podendo passar mensagens com erros.

5.2.1 Verificações de paridade

Adicione mais um bit no final para que o total de bits 1’s seja um número par. Caso o receptor conte um número de bits ímpar, um erro ocorreu.

Em geral, os erros ocorrem em rajadas, então mais de um bit pode estar corrompido.

FIGURA 5.5 PARIDADE PAR BIDIMENSIONAL



Os bits são divididos em linhas e colunas, com um valor de paridade calculado para cada. Dessa forma, verificando uma linha e coluna, identifica-se onde ocorreu um erro e ele pode ser corrigido.

5.2.2 Métodos de soma de verificação

Os bits são tratados como inteiros de 16 bits e são somados. O complemento de 1 da soma forma a soma de verificação da internet e é colocado no cabeçalho do UDP. O receptor faz outro complemento de 1 e verifica se há somente bits 1, se há um 0, então houve um erro.

5.2.3 Códigos de redundância cíclica (CRC)

5.3 Enlaces e protocolos de acesso múltiplo

Enlace de difusão pode possuir diferentes nós remetentes e receptores.

Um nó propaga o quadro e todos recebem uma cópia, gerando um problema para identificar quem fala e quando fala.

Com muitos nós querendo falar ao mesmo tempo, podem ocorrer colisões, onde os pacotes são perdidos e o tempo gasto na difusão é desperdiçado.

5.3.1 Protocolos de divisão de canal

TDM divide o tempo em quadros temporais e aloca compartimentos de tempo para nós diferentes. Cada nó transmite durante seu respectivo compartimento. Embora justo, TDM tem desvantagens, como limitação de velocidade média e necessidade de esperar a vez de transmissão, mesmo quando o nó é o único com pacotes para enviar.

FDM divide o canal em frequências diferentes, reservando cada frequência para um nó, criando canais menores. Ele compartilha as vantagens de TDM, mas também tem a desvantagem de limitar um nó à largura de banda de R/N, mesmo quando é o único com pacotes para enviar.

CDMA (Acesso múltiplo por de código) atribui códigos diferentes a cada nó, permitindo que transmitam simultaneamente. Os códigos exclusivos permitem que os receptores identifiquem corretamente os bits, apesar das interferências. CDMA é utilizado em sistemas militares e civis, sendo especialmente útil em canais sem fio devido às suas propriedades anti-interferências.

5.3.2 Protocolos de acesso aleatório

O dados sempre é transmitido na taxa total do canal.

Cada nó tenta transmitir seus dados, caso ocorra uma colisão, cada um define um tempo e tenta novamente após esse tempo passar.

Slotted ALOHA

- Todos os quadros possuem o mesmo tamanho
- O tempo são intervalos que um quadro possa ser totalmente transmitido
- Os nós começam a enviar dados no início do intervalo.
- Cada nó sabe onde os intervalos começam pois estão sincronizados
- Se houver uma colisão, todos os nós a detectam

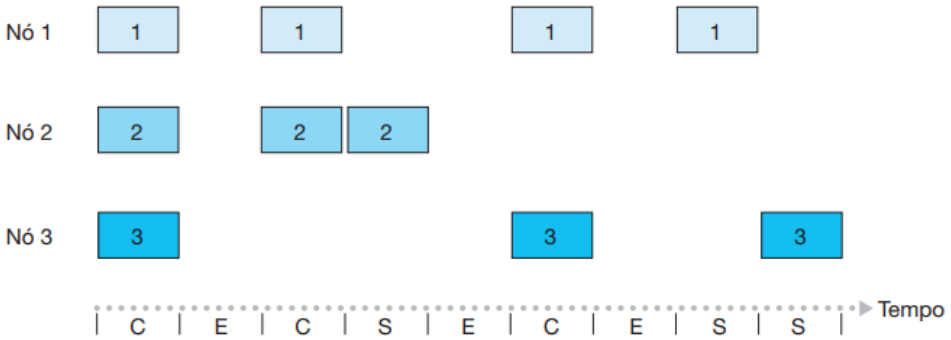
Funcionamento

- Quando o nó tem um novo quadro, espera o próximo intervalo e o envia
- Se foi transmitido com sucesso, não há retransmissão
- Se houver uma colisão, o nó detecta antes do final do intervalo. O quadro é retransmitindo com uma probabilidade p até que ocorra um sucesso

Permite a transmissão com taxa total do canal quando ele é o único ativo. É independente pois cada nó detecta a colisão e decide o que fazer.

Com mais nós, há mais chances de erros e cada colisão é um tempo desperdiçado. Os nós podem detectar colisões mais rapidamente do que enviar dados

FIGURA 5.10 **NÓS 1, 2 E 3 COLIDEM NO PRIMEIRO INTERVALO. O NÓ 2 FINALMENTE É BEM-SUCEDIDO NO QUARTO INTERVALO, O NÓ 1 NO OITAVO INTERVALO E O NÓ 3 NO NONO INTERVALO**



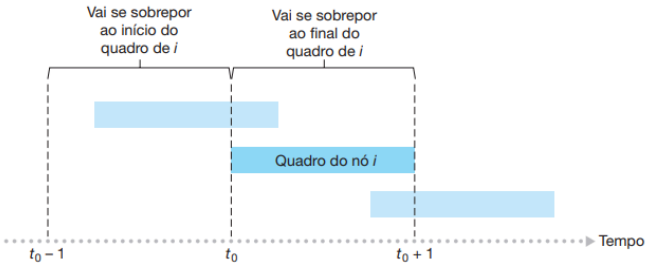
Legenda:
C = Intervalo de colisão
E = Intervalo vazio
S = Intervalo bem-sucedido

ALOHA

Quando ocorre uma colisão, o nó enviará novamente o dado com uma probabilidade p, e vai tentando até que o dado seja de fato enviado.

Se alguém tentar enviar enquanto uma transmissão está terminando, ambos os frames precisam ser retransmitidos.

FIGURA 5.11 TRANSMISSÕES INTERFERENTES NO ALOHA PURO



CSMA (acesso múltiplo com detecção de portadora)

Escuta antes de falar.

Se perceber que o canal está ocioso, transmite o quadro, se percebe que está ocupado, atrasa a transmissão.

Devido ao atraso de propagação, dois nós podem achar que o canal está livre, então podem ocorrer colisões.

CSMA/CD (Colisão de detecção)

Igual o CSMA, porém, quando detecta a colisão, já interrompe a transmissão para evitar desperdício no canal. Além disso, envia um sinal para todos.

Fácil de ser implementado em meios com fio, pois pode-se medir a intensidade do sinal

1. Placa de rede recebe um datagrama da camada de rede e cria um quadro
2. Verifica se o canal está ocioso, se está, envia os dados
3. Se não, espera ele esvaziar para depois enviar
4. Se detecta uma outra transmissão, interrompe o envio
5. Entra backoff de colisão:
    a. Após m colisões, a placa de rede escolhe um K entre 0 até 2^(m-1). Espera-se um tempo de K*512 e volta ao passo 2

Quanto maior o quadro, menor a chance de colisões, pois uma colisão somente poderá ocorrer no seu início. Assim, é possível enviar todos os dados sem muitas dificuldades, porém outros nós esperarão por muito tempo

5.3.3 Protocolos de revezamento MAC

Polling

Um nó mestre convida outros nós a transmitirem em rodadas, escolhendo os outros por alternância circular, perguntando a cada um se possuem algo a ser transmitido, gastando um tempo desnecessário muitas vezes.

Se o mestre falha, o canal fica inoperante.

Passagem de token

Um token é passado de nó para nó. Caso o nó possua esse token, ele pode transmitir os dados, depois passa o token para frente.

Se a máquina que possui o token saiu, é então usado um processo que gera um novo token.

5.4 Redes locais comutadas

5.4.1 Endereçamento MAC e ARP

Endereço MAC

O endereço MAC possui 48 bits em hexadecimal e é queimado na placa NIC, podendo ser modificada por software

Leva quadros de uma interface para outra que esteja conectada na mesma rede.

Cada MAC é único. Quando é necessário, é comprado na IEEE um bloco de 24 bits (os mais da esquerda), e podem ser usadas as outras combinações de bits

Um switch com máquinas conectadas poderá transmitir dados entre elas utilizando os endereços MAC. Se um quadro chegou e o endereço MAC é diferente, então esta placa de rede ignora o quadro.

ARP (Protocolo de resolução de endereços)

Converte um endereço IP em um endereço MAC.

Cada nó possui uma tabela ARP com endereços IP mapeados para MAC e um tempo de vida para essa tradução, já que um dispositivo pode sair da rede, assim o IP que é limitado poderia se esgotar com o tempo caso os registros não fossem removidos.

A quer enviar um datagrama para B:

- endereço de B não está na tabela ARP de A
- A envia um broadcast com o IP de B para todos os nós na LAN, que recebem a requisição ARP
- B recebe e responde para A com o seu endereço MAC
- A salva o endereço em sua tabela ARP até que a informação expire

5.4.2 Ethernet

Cabos de par trançado conectados a hubs, gerando uma topologia do tipo estrela, possuindo ainda colisões na rede, pois os dados são replicados para todos os nós da rede.

Depois foram substituídos por switches que evitam colisões e aumentam a vazão da rede

Estrutura do quadro Ethernet

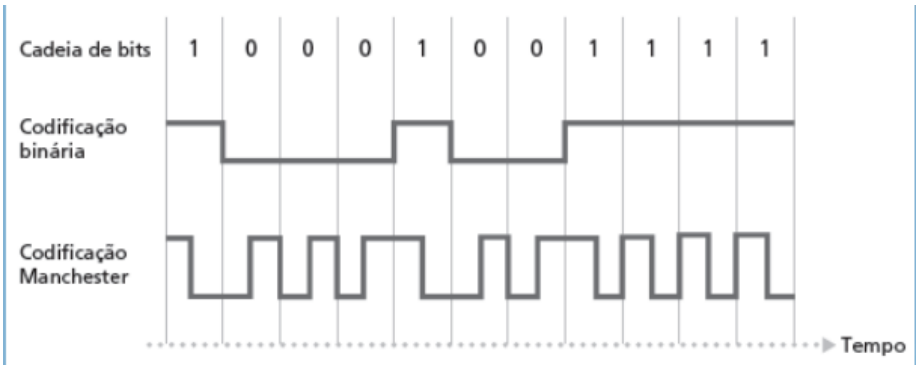
FIGURA 5.20 ESTRUTURA DO QUADRO ETHERNET

- O preâmbulo é um conjunto de 8 bytes, sendo 7 com o padrão 10101010 e outro com padrão 10101011 para sincronizar as taxas de clock entre emissor e receptor
- Endereços MAC com 6 bytes para destino e origem. O receptor verifica se o destino é o seu MAC e, se for, desencapsula o pacote passa para a camada de rede.
- O tipo indica o protocolo da camada de cima que irá receber os dados.

- CRC: Permite que o receptor detecte que ocorreu algum erro e jogue fora os dados

Se os dados não atingirem um tamanho mínimo, então é adicionado um enchimento dos dados para evitar colisões

Codificação Manchester



Permite que os clocks sejam sincronizados

Um bit 0 é uma passagem de zero para um em um intervalo de tempo e um bit 1 é uma passagem de 1 para 0

Ethernet: não confiável, sem conexão

Não há uma apresentação entre emissor e receptor, os dados são simplesmente enviados

Não é confiável, pois não há a confirmação de que os dados chegaram ou não. Cabe ao TCP recuperar um dado perdido.

A ethernet usa o CSMA/CD não slotted.

5.4.3 Comutadores da camada de enlace

Hubs

Recebem um pacote e o retransmite para todas as suas portas, podendo gerar colisões já que não possuem um buffer para armazenar o que chega.

As placas de rede dos hospedeiros é quem determina se houve ou não colisão.

Switch

Armazena e repassa os quadros ethernet.

Examina o endereço MAC do quadro que chega e determina para onde o pacote deve ir

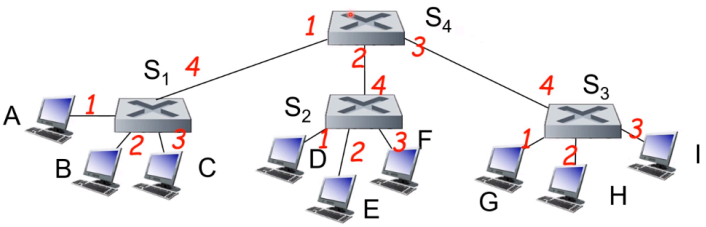
Os hospedeiros não sabem da presença dos switch, que por sua vez aprendem de forma autônoma a montar a sua tabela de comutação com a informação do endereço MAC do hospedeiro.

.Podem ocorrer transmissões simultâneas pois não permite que ocorram colisões, ainda mais por serem full-duplex, tendo um cabo para enviar e outro para receber.

Quando recebe um quadro, verifica se já existe o endereço do transmissor na tabela de comutação, senão, adiciona o seu MAC, sua interface e o TTL.

Se o destino não é conhecido, é feito um broadcast. Se conhece, é feito um envio seletivo.

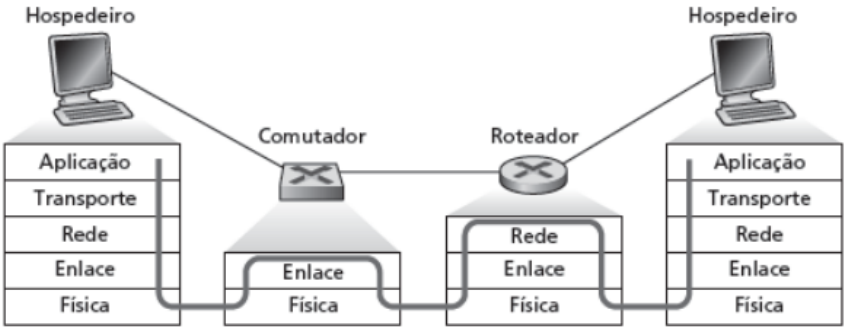
Switches podem ser interconectados, mas funcionam da mesma forma



Comutadores versus roteadores

Roteadores: Examina cabeçalhos da camada de rede e computam tabelas usando algoritmos de roteamento e endereços IP

Comutadores (Switches): examinam o cabeçalho da camada de enlace. Aprendem a tabela por broadcasting ou recebimento de pacotes.



5.4.4 Redes locais virtuais (VLANs)

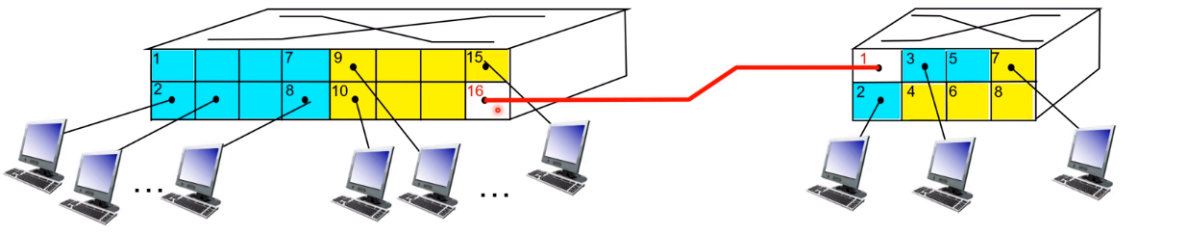
Comutadores que podem criar diferentes LANs a partir de uma única estrutura física

As VLANs baseadas em portas agrupam determinadas portas de um comutador para que ele funcione como múltiplos comutadores (switches).

Portas de 1 para 8 só alcançam essas portas, sendo que podem ser modificadas dinamicamente.

As LANs podem ser definidas com base no endereço MAC das extremidades em vez da porta do comutador.

Uma porta pode ser usada para conectar diferentes switches a partir de uma porta de tronco, podendo carregar os quadros com as informações de cada VLAN. Amarelo só vai se comunicar com amarelo e azul com azul. Porém devem usar o quadro 802.1Q, um pouco diferente do ethernet normal, pois ele contém uma tag para o rótulo que deve ser enviado os dados

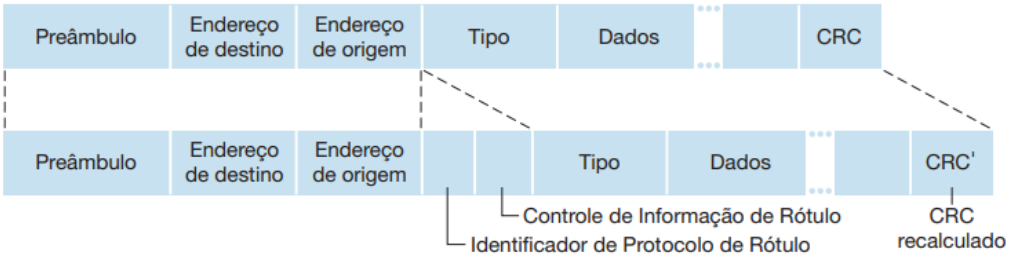


Engenharia Elétrica
(portas VLAN 1-8)

Ciência da Computação
(portas VLAN 9-15)

Portas 2,3,5 pertecem a VLAN da EE
Portas 4,6,7,8 pertencem a VLAN da CS

FIGURA 5.27 QUADRO ETHERNET ORIGINAL (NO ALTO); QUADRO VLAN ETHERNET 802.1Q-TAGGED (EMBAIXO)



5.7.1 Um dia na vida de uma solicitação de página web

Cenário: Um aluno conecta o laptop à rede do campus e solicita/recebe ww.google.com

Primeiramente, o laptop precisa obter um IP. É feito um pacote DHCP com UDP, IP e ethernet.

O quadro é enviado por broadcasting para a LAN e espera-se que alguém seja um servidor DHCP.

O servidor DHCP formula um DHCP ACK, com IP do cliente, IP do primeiro salto, nome e endereço IP do servidor DNS. O pacote vira um quadro e vai diretamente para o laptop, pois o switch já aprendeu o caminho.

O quadro é desencapsulado e o cliente já tem o seu IP, nome e endereço do servidor DNS e já sabe o endereço do roteador de primeiro salto (gateway)

Agora é necessário saber o IP do google.com. para isso, é criada uma requisição DNS encapsulada em UDP, encapsulada em IP e encapsulada em ethernet.

Porém é necessário saber o MAC do roteador de primeiro salto, então é feito uma requisição ARP por broadcast, recebida pelo roteador que responde com o ARC reply, dando o endereço MAC da interface do roteador.

Então é enviado um quadro Ethernet para consulta de DNS ao endereço MAC do roteador de borda.

Então o laptop envia um quadro para o gateway, que por sua vez envia para o servidor DNS, sendo roteado por meio das tabelas de roteamento.

Lá ele é demultiplexado e o servidor DNS responde com o endereço IP do google.

Agora o cliente monta uma mensagem HTTP, com TCP, que precisa abrir uma conexão, que vai até o servidor do google e abre a conexão, enviando um SYN ACK e a conexão está estabelecida.

O HTTP é novamente encapsulado no TCP, IP, Ethernet e passa pela rede da universidade até o servidor do google onde será lida e aberta pelo servidor web.

O servidor web faz a resposta HTTP, encapsula ela e encaminha na rede para que chegue até o laptop do cliente, que desencapsula e possui a página desejada.

Capítulo 6 - Redes sem fio e redes móveis

Elementos de uma rede sem fio

São hospedeiros que executam as aplicações e podem ser móveis ou estáticos

Estação base são aquelas que conectam o host à internet. Geralmente ela está conectada com fio. Mas pode estar conectada sem fio também. Podem receber e enviar pacotes para os hospedeiros

O enlace sem fio conecta os dispositivos à estação base e usam o ar .

Modo infraestrutura: as estações-base conectam os hospedeiros móveis à rede com fio e o hospedeiro pode trocar de estação-base sem perder conexão. Wifis bem estabelecidos em uma rede, mantendo o mesmo IP.

Modo ad hoc: Os dispositivos se auto organizam, roteando dados entre eles mesmos, como se fosse o bluetooth. Possui um nó mestre que conecta a conexão.

Taxonomia de redes sem fio

Único salto com infraestrutura: Hospedeiro conecta a uma estação base conectada a uma rede sem fio

Múltiplos saltos com infraestrutura: Um hospedeiro transmite para um sensor, que depois transmite para uma estação-base

Único salto sem infraestrutura: Conexões bluetooth e redes ad hoc. Teclados sem fio, por exemplo

Múltiplos saltos sem infraestrutura: Não possui infraestrutura nem conexão na internet. Veículos autônomos que se aproximam podem se comunicar para falarem o que irão fazer.

6.2 Características de enlaces e redes sem fio

Diferenças do enlace com fio:

- O sinal se perde mais rapidamente do que em enlaces com fio
- Como as frequências são padronizadas, interferências são muito comuns

- O sinal de rádio reflete em diferentes objetos, chegando em tempos diferentes no destino

A razão sinal-ruído (SNR) é a razão da quantidade de sinais sobre a quantidade de ruído. Dessa forma, quanto maior a razão, melhor, pois teremos mais sinal do que ruído. Porém, quanto maior a potência para aumentar o sinal, maior o gasto de energia, ou seja, a bateria dura menos, ademais os sinais podem se interferir ainda mais.

Quanto maior a frequência, maior a velocidade de transmissão, menor a distância e menor a taxa de sucesso.

Pode ocorrer a presença de obstáculos ou simplesmente uma atenuação do sinal de forma que A escuta B, B escuta A e C, mas A e C não se escutam. Então se A e C decidem transmitir, pois não foi detectado nenhum sinal no meio, poderá ocorrer uma colisão em B, que recebe dos dois.

6.2.1 CDMA

Todos os usuários compartilham a mesma frequência, mas possuem um código para codificar os seus dados. Assim, eles podem existir ao mesmo tempo com pouca interferência.

Um receptor que recebe a transmissão de vários transmissores consegue pegar o sinal resultante e verificar o que cada um transmite

sinal codificado = dados originais * sequência de chipping

decodificação: produto entre sinal codificado e sequência de chipping

6.3 WiFi: Lans sem fio 802.11

Possui diferentes padrões, mas todos usam o CSMA/CA para evitar colisões e todos possuem versões de estação-base e rede ad-hoc. Além de possuírem o mesmo tamanho de quadro.

Diferem-se na faixa de frequência, largura do canal, forma de codificação, etc.

TABELA 6.1 RESUMO DOS PADRÕES IEEE 802.11

Padrão	Faixa de frequências (EUA)	Taxa de dados
802.11b	2,4–2,485 GHz	até 11 Mbits/s
802.11a	5,1–5,8 GHz	até 54 Mbits/s
802.11g	2,4–2,485 GHz	até 54 Mbits/s

O 802.11n possui múltiplas antenas e funciona na faixa de 2,4 a 2,5GHz e pode chegar até 600Mbps, muito difícil de chegar devido à sobrecarga de protocolos.

6.3.1 A arquitetura 802.11

Hospedeiro sem fio se comunica com a estação-base (Ponto de acesso), formando uma célula, ou seja, uma área onde o ponto de acesso atua. Pode ser usado o modo ad hoc somente para os hospedeiros

O 802.11b é dividido em 11 canais em diferentes frequências, sendo que cabe ao administrador do ponto de acesso escolher as frequências, sendo que os canais 1, 6 e 11 não se sobrepõem. Caso um ponto de acesso vizinho tenha o mesmo canal, pode ocorrer interferência e atrapalhar as duas células.

Um hospedeiro precisa se associar a um ponto de acesso. Então ele começa fazendo uma varredura nos canais e escuta os beacons, quadro de sinalização com o endereço MAC. O hospedeiro então escolhe qual ele quer se conectar e pode realizar a autenticação. Depois de conectado, ele obterá um IP por meio do DHCP na sub-rede do ponto de acesso.

Varredura passiva: O cliente espera o ponto de acesso sinalizar a sua existência, escolhe um, envia um quadro de solicitação de associação e recebe uma resposta

Varredura ativa: O hospedeiro manda uma mensagem para todos os pontos de acesso que estão ao seu alcance, que o responde com um quadro de resposta.

6.3.2 O protocolo MAC 802.11

Evita colisões entre 2 ou mais nós que transmitem informações ao mesmo tempo, pois utiliza o CSMA

Não possui a detecção de colisão pois há uma grande dificuldade de verificar se algo está sendo transmitido, já que o transmissor está junto com o receptor, o próprio dado que está sendo transmitido é escutado de uma forma muito mais forte.

Utiliza o **CSMA/CA**, que evita colisões, mas não as detecta

Se o transmissor percebe que o canal está ocioso por um certo período de tempo, DIFS, ele transmite todo o quadro.

Se percebe que o canal está ocupado, ele inicia um tempo de espera que vai ser definido de forma aleatória, então inicia um temporizador que fica travado até o momento em que o canal fica ocioso. Depois que ele expira, os dados são transmitidos e espera-se que o receptor mande um ACK após um curto período de tempo interquadro (SIFS). Se não recebe, aumenta o intervalo de espera.

Para o envio de quadros mais longos, primeiro o transmissor envia um pacote pequeno solicitando uma reserva do canal (Request to send - RTS), que pode colidir, mas devido ao seu tamanho o prejuízo é pequeno. Depois o receptor envia uma mensagem clear to send (CTS) por broadcast. Dessa forma, todos os nós escutam e param de transmitir, liberando espaço para que o pacote grande seja enviado sem problemas de colisão. Nem todos os roteadores possuem essa capacidade. Alguns que possuem, permitem determinar um tamanho mínimo do quadro para que ocorra a transmissão to RTS e CTS

6.3.3 O quadro IEEE 802.11

2	2	6	6	6	2	6	0 - 2312	4
quadro de controle	duração	endereço 1	endereço 2	endereço 3	controle de seq.	endereço 4	carga	CRC

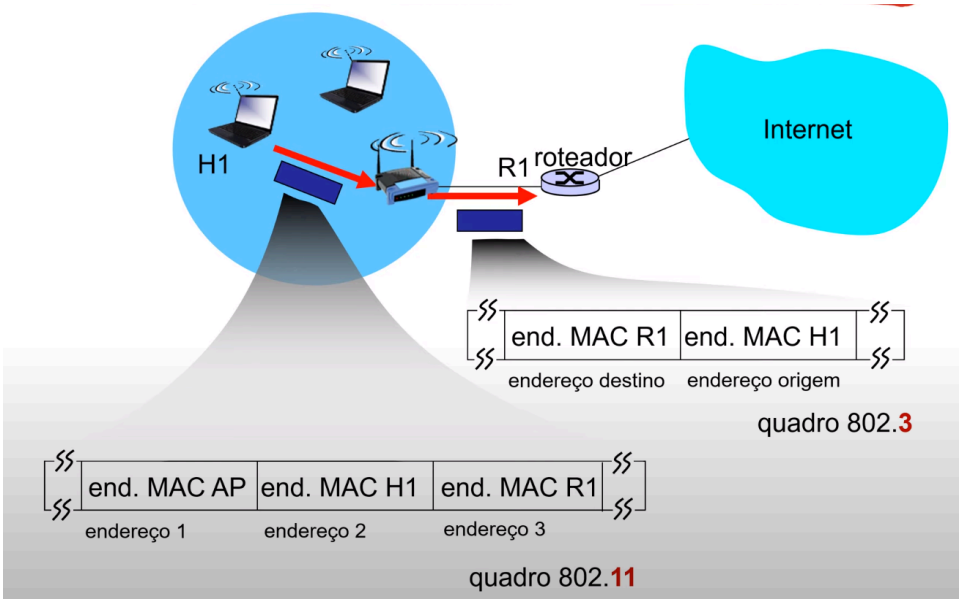
Função	ToDS (para AP)	FromDS (de AP)	Endereço 1 (receptor)	Endereço 2 (transmissor)	Endereço 3	Endereço 4
Redes Ad-hoc	0	0	End. Destinatário	End. Origem	BSSID	Não Usado
Quadro enviado para AP (infra.)	1	0	BSSID	End. Origem	End. Destinatário	Não Usado
Quadro enviado pelo AP (infra.)	0	1	End. Destinatário	BSSID	End. Origem	Não Usado
WDS (ponte)	1	1	End. Receptor (próxima estação que recebe o quadro)	End. Transmissor (MAC da estação que transmitiu o quadro)	End. Destinatário	End. Origem

BSSID (Identificador de BSS) identifica unicamente cada BSS. Quando o quadro é vindo de uma estação que opera em modo infra-estrutura BSS, BSSID é o endereço MAC do AP. Quando o quadro é vindo de uma estação que opera em modo *ad hoc* (IBSS), o BSSID é um número aleatório gerado e localmente administrado pela estação que iniciou a transmissão.

O campo duração serve saber quanto tempo será a duração do envio de uma solicitação de espera (RTS/CTS)

O controle de sequência permite saber se os pacotes estão chegando em ordem, igual ao TCP.

Exemplo de transmissão de quadro de uma máquina para a internet usando a segunda linha da tabela acima

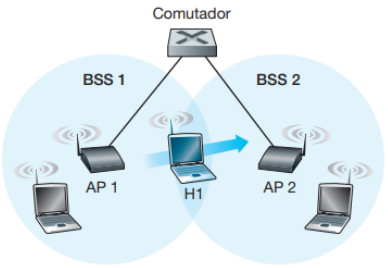


6.3.4 Mobilidade em uma sub-rede

O IP mudará somente o ponto de acesso, o IP continuará o mesmo.

O novo ponto de acesso, ao identificar uma conexão, envia um sinal broadcast para o switch, que então perceberá que o hospedeiro mudou de local, então o seu ponto de acesso deverá ser atualizado na tabela, então novos pacotes serão enviados para o novo ponto de acesso.

FIGURA 6.15 MOBILIDADE NA MESMA SUB-REDE



6.4 Acesso celular à internet

Cada célula possui uma estação base

Cada estação-base está conectada a uma rede de computação móvel, sendo responsáveis para configurações de chamadas e permitindo a mobilidade entre as diferentes estações base

6.4.1 Visão geral da arquitetura de rede celular

Duas técnicas para combinar o espectro de rádio da estação móvel para a estação-base:

FDMA/TDMA combinado: Divide o espectro em frequência e, em cada frequência, divide ela por tempo

CDMA como já explicado anteriormente