**Capitulo 5 Resumo camada de Redes**

A camada de rede é responsável pela transferência de pacotes da origem ao destino, frequentemente passando por vários roteadores intermediários. Diferente da camada de enlace de dados, que apenas move quadros de uma extremidade a outra de um fio, a camada de rede lida com a transmissão ponto a ponto. Para isso, precisa conhecer a topologia da rede e escolher os melhores caminhos, evitando sobrecarregar algumas rotas enquanto outras permanecem ociosas. Quando a origem e o destino estão em redes diferentes, surgem novos desafios que também são responsabilidade da camada de rede. Este capítulo aborda essas questões, com foco na Internet e no protocolo IP.

**5. Questões de Projeto da Camada de Rede**

Neste capítulo, são apresentadas as principais questões que os projetistas da camada de rede devem considerar, focando principalmente no serviço oferecido à camada de transporte e no projeto interno da rede.

**5.1.1 Comutação de Pacotes Store-and-Forward**

**Contexto Operacional dos Protocolos da Camada de Rede**

Antes de aprofundar nos detalhes da camada de rede, é essencial entender o contexto em que seus protocolos operam. A Figura 5.1 ilustra esse contexto, destacando os principais componentes de rede: os equipamentos do ISP (roteadores conectados por linhas de transmissão) e os equipamentos dos clientes.

1. **Componentes de Rede:**
   * **Equipamentos do ISP:** Incluem roteadores interconectados por linhas de transmissão, representados dentro da elipse sombreada.
   * **Equipamentos dos Clientes:** Incluem dispositivos como computadores pessoais e roteadores pertencentes aos clientes, representados fora da elipse.
2. **Exemplos de Conexões:**
   * **Host H1:** Diretamente conectado a um roteador do ISP (A), similar a um computador pessoal conectado a um modem DSL.
   * **Host H2:** Localizado em uma LAN, como a Ethernet de um escritório, conectado a um roteador do cliente (F), que também se conecta ao equipamento do ISP por uma linha dedicada.

**Função dos Roteadores**

Apesar de F não pertencer ao ISP, ele é tratado como parte da rede do ISP para fins deste capítulo, já que executa os mesmos algoritmos que os roteadores do ISP. A principal preocupação é estudar esses algoritmos.

**Processo de Comutação de Pacotes Store-and-Forward**

O processo de comutação de pacotes store-and-forward opera da seguinte forma:

1. **Transmissão do Pacote:**
   * Um host envia um pacote para o roteador mais próximo, seja dentro de sua LAN ou através de um enlace ponto a ponto para o ISP.
2. **Armazenamento e Verificação:**
   * O pacote é armazenado no roteador até ser completamente recebido e verificado (checksum).
3. **Encaminhamento:**
   * Após a verificação, o pacote é encaminhado ao próximo roteador ao longo do caminho até chegar ao host de destino.

Este mecanismo é central para a operação da camada de rede e foi discutido em capítulos anteriores.

**5.1.2 Serviços Oferecidos à Camada de Transporte**

A camada de rede oferece serviços à camada de transporte na interface entre ambas, sendo crucial determinar quais serviços são necessários. Esses serviços devem atender a três objetivos principais:

1. **Independência Tecnológica:** Os serviços devem funcionar independentemente da tecnologia usada nos roteadores.
2. **Isolamento da Camada de Transporte:** A camada de transporte deve estar isolada em termos de número, tipo e topologia dos roteadores.
3. **Uniformidade de Endereçamento:** Os endereços de rede devem seguir um plano de numeração uniforme, aplicável tanto em LANs quanto em WANs.

**Liberdade de Especificação dos Serviços**

Com esses objetivos em mente, os projetistas da camada de rede têm liberdade para especificar os serviços oferecidos. Essa liberdade leva a um debate intenso sobre se a camada de rede deve ser orientada a conexões ou não.

1. **Serviço Não Orientado a Conexões:**
   * **Comunidade da Internet:** Argumenta que a rede deve apenas movimentar pacotes, já que é inerentemente não confiável. Assim, os hosts devem lidar com controle de erros e controle de fluxo.
   * **Princípio de Projeto Fim a Fim:** Os serviços de rede devem ser simples, como **SEND PACKET** e **RECEIVE PACKET**, sem ordenação de pacotes ou controle de fluxo, pois os hosts já realizam essas funções.
   * **Endereçamento Completo:** Cada pacote deve incluir o endereço de destino completo, sendo transportado independentemente de outros pacotes.
2. **Serviço Orientado a Conexões:**
   * **Companhias Telefônicas:** Defendem um serviço de rede confiável e orientado a conexões, argumentando que a qualidade de serviço (QoS) é crucial, especialmente para tráfego em tempo real como voz e vídeo.
   * **Histórico Telefônico:** Baseiam-se na experiência de cem anos com sistemas telefônicos, onde a QoS é primordial e a ausência de conexões torna difícil manter essa qualidade.

**Evolução das Redes**

O debate entre serviços orientados e não orientados a conexões continua vivo. Historicamente, redes como X.25 e Frame Relay eram orientadas a conexões, enquanto a Internet e a ARPANET adotaram redes não orientadas a conexões, com o IP se tornando amplamente popular.

* **Protocolo IP:** Tornou-se o padrão, apesar da concorrência de tecnologias orientadas a conexões como ATM (Asynchronous Transfer Mode).
* **Evolução para Conexões:** A Internet está evoluindo para incorporar características orientadas a conexões, como QoS. Tecnologias como MPLS (MultiProtocol Label Switching) e VLANs são exemplos de abordagens orientadas a conexões amplamente utilizadas.

**5.1.3 Implementação do Serviço Não Orientado a Conexões**

**Conceito de Redes de Datagramas**

Quando a camada de rede oferece um serviço não orientado a conexões, os pacotes são enviados de forma independente, sem a necessidade de configuração antecipada. Esses pacotes, chamados datagramas, são injetados na rede e roteados individualmente, formando o que é conhecido como rede de datagramas.

**Funcionamento de uma Rede de Datagramas**

1. **Transmissão de Mensagens:**
   * O processo de envio começa com o processo P1 em H1, que quer enviar uma mensagem longa para P2 em H2.
   * P1 entrega a mensagem à camada de transporte, que adiciona um cabeçalho de transporte e passa a mensagem para a camada de rede.
2. **Divisão da Mensagem em Pacotes:**
   * Se a mensagem é maior que o tamanho máximo de pacote, a camada de rede a divide em pacotes menores.
   * No exemplo, a mensagem é dividida em quatro pacotes (1, 2, 3 e 4).
3. **Encaminhamento dos Pacotes:**
   * Cada pacote é enviado ao roteador A via um protocolo ponto a ponto, como o PPP.
   * O roteador A usa sua tabela de roteamento para decidir para onde enviar cada pacote. Esta tabela contém destinos e as interfaces de saída correspondentes.
4. **Processo de Roteamento:**
   * Pacotes 1, 2 e 3 são armazenados em A, seus checksums são verificados, e são encaminhados para o roteador C.
   * De C, os pacotes seguem para E, depois para F, e finalmente chegam a H2 via LAN.
5. **Roteamento Diferente para o Pacote 4:**
   * O pacote 4, ao chegar ao roteador A, é enviado para B em vez de seguir a rota dos outros pacotes.
   * Isso pode ocorrer devido a uma atualização na tabela de roteamento de A, talvez por uma obstrução de tráfego na rota ACE.
   * A tabela de roteamento de A é atualizada conforme necessário, um processo gerido pelo algoritmo de roteamento.

**Importância do Algoritmo de Roteamento**

O algoritmo de roteamento é fundamental para a gestão das tabelas de roteamento e para a tomada de decisões sobre o encaminhamento de pacotes. Existem diferentes tipos de algoritmos de roteamento, que serão discutidos em detalhes ao longo do capítulo.

**Exemplo do Protocolo IP**

O IP (Internet Protocol) é um exemplo proeminente de serviço de rede não orientado a conexões:

* **Endereçamento:** Cada pacote IP contém um endereço de destino, usado pelos roteadores para encaminhá-lo individualmente.
* **IPv4 e IPv6:** Os endereços têm 32 bits no IPv4 e 128 bits no IPv6.
* **Detalhamento:** O IP será detalhado mais adiante no capítulo.

**Implementação do Serviço Orientado a Conexões**

**Conceito de Redes de Circuitos Virtuais**

Para o serviço orientado a conexões, utilizamos redes de circuitos virtuais. A principal vantagem é evitar a escolha de novas rotas para cada pacote enviado. Em vez disso, uma rota é estabelecida desde a origem até o destino durante a configuração da conexão, sendo armazenada nas tabelas dos roteadores. Todos os pacotes seguem essa rota até a conexão ser liberada, momento em que o circuito virtual é encerrado.

**Funcionamento de uma Rede de Circuitos Virtuais**

1. **Estabelecimento da Conexão:**
   * Quando um host deseja estabelecer uma conexão, uma rota é selecionada e configurada em cada roteador ao longo do caminho.
   * A conexão é identificada por um identificador de circuito virtual, incluído em cada pacote pertencente a essa conexão.
2. **Encaminhamento dos Pacotes:**
   * A rota é armazenada em tabelas de roteamento nos roteadores. Cada entrada da tabela indica a interface de saída e o identificador de conexão a ser usado.
   * Os pacotes são encaminhados ao longo da rota pré-estabelecida, mantendo o identificador de conexão.
3. **Exemplo de Roteamento:**
   * Considere o host H1 estabelecendo uma conexão com H2. A tabela de roteamento em cada roteador ao longo da rota (A, C, E) registra a conexão.
   * A primeira entrada na tabela de A indica que pacotes com o identificador de conexão 1 são enviados para C, mantendo o mesmo identificador.
   * Similarmente, a tabela de C encaminha esses pacotes para E, também com o identificador de conexão 1.
4. **Estabelecimento de Múltiplas Conexões:**
   * Se H3 desejar estabelecer uma conexão com H2 usando o mesmo identificador de conexão (1), ocorre um conflito.
   * Para resolver isso, A atribui um novo identificador de conexão para o tráfego de saída correspondente à segunda conexão, evitando confusões em C.
   * Essa troca de identificadores é essencial para distinguir diferentes conexões.

**Troca de Rótulos**

A troca de identificadores de conexão é conhecida como troca de rótulos. Um exemplo de serviço de rede orientado a conexões que utiliza esse método é o **MPLS (MultiProtocol Label Switching)**.

* **MPLS:** Utilizado principalmente nas redes dos ISPs, encapsulando pacotes IP em cabeçalhos MPLS com um rótulo de conexão de 20 bits.
* **Funcionalidade Oculta:** O MPLS opera de maneira transparente para os clientes, sendo gerido pelo ISP para otimizar o tráfego e melhorar a qualidade de serviço.
* **Qualidade de Serviço (QoS):** O MPLS ajuda a garantir QoS e facilita o gerenciamento eficiente do tráfego.

**5.2 Algoritmos de Roteamento**

**Função Principal da Camada de Rede**

A função principal da camada de rede é rotear pacotes da máquina de origem até a máquina de destino, geralmente através de múltiplos hops. Os algoritmos de roteamento determinam a rota ideal para os pacotes, garantindo que eles alcancem seu destino de maneira eficiente e confiável.

**Roteamento em Redes de Datagramas vs. Circuitos Virtuais**

* **Redes de Datagramas:** Cada pacote é roteado independentemente, e as decisões de roteamento são tomadas várias vezes ao longo do trajeto.
* **Redes de Circuitos Virtuais:** A rota é determinada quando a conexão é estabelecida, e todos os pacotes seguem essa rota predefinida até a conexão ser encerrada.

**Diferença entre Roteamento e Encaminhamento**

* **Roteamento:** Processo de decisão sobre quais rotas utilizar. O algoritmo de roteamento é responsável por preencher e atualizar as tabelas de roteamento.
* **Encaminhamento:** Processo de direcionamento de pacotes que chegam a uma interface de saída específica, conforme indicado pelas tabelas de roteamento.

**Propriedades Desejáveis em um Algoritmo de Roteamento**

1. **Exatidão:** O algoritmo deve sempre encontrar a rota correta.
2. **Simplicidade:** O algoritmo deve ser fácil de implementar e entender.
3. **Robustez:** Deve lidar bem com falhas de hardware e software, além de mudanças na topologia.
4. **Estabilidade:** Deve alcançar rapidamente um ponto de equilíbrio após uma mudança.
5. **Equidade:** Todas as conexões devem ser tratadas de forma justa.
6. **Eficiência:** Maximização do throughput e minimização do atraso médio dos pacotes.

**Algoritmos de Roteamento Não Adaptativos vs. Adaptativos**

* **Algoritmos Não Adaptativos (Estáticos):**
  + Não mudam em resposta a mudanças na topologia ou no tráfego.
  + As rotas são pré-calculadas e carregadas nos roteadores quando a rede é iniciada.
  + Úteis quando a escolha de rotas é óbvia e a rede é relativamente estável.
* **Algoritmos Adaptativos (Dinâmicos):**
  + Ajustam as decisões de roteamento com base em mudanças na topologia e no tráfego.
  + Obtêm informações em diferentes momentos (quando a topologia muda, a cada intervalo de tempo, etc.) e de diferentes fontes (local, roteadores adjacentes, todos os roteadores).
  + Utilizam diferentes métricas de otimização (distância, número de hops, tempo estimado de tráfego).

**Comparação entre Redes de Circuitos Virtuais e de Datagramas**

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

**Conflito entre Equidade e Eficiência**

A eficiência máxima do fluxo total pode, às vezes, entrar em conflito com a equidade das conexões individuais. Por exemplo, para maximizar o fluxo total em uma rede, pode ser necessário desativar completamente o tráfego entre certos pontos, o que pode ser visto como injusto por aqueles pontos.

**Otimização no Roteamento**

Os objetivos comuns na otimização de algoritmos de roteamento incluem a minimização do atraso médio dos pacotes e a maximização do throughput total da rede. Muitas redes tentam minimizar a distância que um pacote deve percorrer ou o número de hops, melhorando o atraso e a largura de banda consumida.

**5.2.1Princípio de Otimização**

**Descrição Geral**

O princípio de otimização, proposto por Bellman em 1957, oferece uma forma de descrever as rotas ideais na rede sem considerar explicitamente a topologia ou o tráfego. Esse princípio estabelece que:

Se o roteador J estiver no caminho ideal entre o roteador I e o roteador K, então o caminho ideal de J até K também estará na mesma rota.

**Justificativa do Princípio**

Para entender melhor, considere:

* r1 como a parte da rota entre I e J.
* r2 como a parte da rota entre J e K.

Se houvesse uma rota melhor que r2 entre J e K, essa nova rota poderia ser concatenada com r1 para formar uma rota mais eficiente entre I e K, contradizendo a premissa de que r1r2 é a rota ideal. Portanto, o caminho ideal de I a K passa por J e utiliza a rota ideal de J até K.

**Árvore de Escoamento**

Uma consequência do princípio de otimização é que as rotas ideais de todas as origens para um determinado destino formam uma estrutura de árvore, chamada de **árvore de escoamento**. A Figura 5.5(b) ilustra uma árvore de escoamento usando o número de hops como métrica de distância.

**Características das Árvores de Escoamento**

1. **Unicidade:** Embora a árvore de escoamento represente uma rota ideal, ela não é necessariamente única. Podem existir outras árvores com caminhos de tamanhos equivalentes.
2. **DAG:** Se considerarmos todos os caminhos possíveis, a estrutura resultante é um **Grafo Acíclico Dirigido** (DAG), que não contém loops. Para simplificação, árvores de escoamento são usadas para representar ambos os casos.
3. **Convergência:** Como árvores de escoamento são estruturas acíclicas, os pacotes sempre serão entregues em um número finito e limitado de hops.

**Considerações Práticas**

Na prática, vários desafios surgem:

* **Falhas de Enlaces e Roteadores:** Enlaces e roteadores podem falhar e voltar a operar, levando a diferentes roteadores a terem visões distintas da topologia atual.
* **Obtenção de Informações:** Surge a questão de como os roteadores obtêm informações sobre a topologia da rede. Cada roteador pode calcular sua própria árvore de escoamento ou essas informações podem ser disseminadas por outros meios.

**Implementação**

O princípio de otimização e as árvores de escoamento servem como benchmarks para avaliar outros algoritmos de roteamento. Esses conceitos são essenciais para a criação de algoritmos que maximizem a eficiência da rede, minimizem o atraso e garantam a entrega confiável dos pacotes.

**5.2.3 Roteamento pelo Caminho Mais Curto**

**Introdução**

O roteamento pelo caminho mais curto é uma técnica fundamental para calcular as rotas ideais em uma rede. Essa técnica se baseia em um grafo que representa a rede, onde cada nó corresponde a um roteador e cada arco a uma interface de comunicação. O algoritmo encontra o caminho mais curto entre dois roteadores no grafo, usando diferentes métricas para determinar o "comprimento" do caminho.

**Métricas de Caminho Mais Curto**

* **Número de Hops:** Conta o número de saltos de um roteador para outro.
* **Distância Geográfica:** Mede a distância física entre os roteadores.
* **Atraso Médio:** Calcula o atraso médio de enfileiramento e transmissão.
* **Largura de Banda:** Considera a capacidade de transmissão dos enlaces.
* **Custo de Comunicação:** Baseado em fatores econômicos ou de recursos.

A escolha da métrica depende dos objetivos específicos da rede, como minimizar o atraso ou maximizar o throughput.

**Algoritmo de Dijkstra**

O algoritmo de Dijkstra é um método bem conhecido para encontrar o caminho mais curto de um nó de origem a todos os destinos em um grafo. Ele funciona da seguinte maneira:

1. **Inicialização:**
   * Rotule o nó de origem com distância zero e todos os outros nós com infinito.
   * Marque o nó de origem como permanente.
2. **Iteração:**
   * Para o nó ativo (inicialmente o nó de origem), examine todos os nós adjacentes.
   * Atualize os rótulos dos nós adjacentes com a soma do rótulo do nó ativo e a distância até o nó adjacente.
   * Se o novo rótulo for menor que o rótulo existente, substitua-o.
   * Marque o nó provisório com o menor rótulo como permanente e torne-o o nó ativo na próxima iteração.
3. **Finalização:**
   * Repita o processo até que todos os nós estejam marcados como permanentes.

**Exemplo de Execução**

Considere o grafo na Figura 5.6(a) onde queremos encontrar o caminho mais curto de A até D.

**Etapa 1:**

* + Marque A como permanente.
  + Rotule B e G com as distâncias de A (B: 2, G: 6).

**Etapa 2:**

* + Marque B como permanente (menor rótulo).
  + Atualize os rótulos de E (4) e C (9) através de B.

**Etapa 3:**

* + Marque E como permanente.
  + Atualize os rótulos de G (5) e F (6) através de E.

**Etapa 4:**

* + Marque G como permanente.
  + Atualize o rótulo de H (9) através de G.

**Etapa 5:**

* + Marque F como permanente.
  + Atualize o rótulo de H (8) através de F.

**Etapa 6:**

* + Marque D como permanente. Chegamos ao destino.

**5.2.3 Flooding**

A inundação, ou flooding, é uma técnica simples e local usada em algoritmos de roteamento, onde cada pacote de entrada é enviado para todas as interfaces de saída, exceto aquela pela qual chegou. Este método garante que um pacote alcance todos os nós da rede, embora gere um grande número de pacotes duplicados. Sem mecanismos de controle, isso poderia resultar em um número infinito de duplicatas de pacotes.

Para mitigar isso, uma abordagem é incluir um contador de saltos no cabeçalho do pacote. O contador é decrementado em cada salto, e o pacote é descartado quando o contador atinge zero. Idealmente, o contador de saltos é ajustado para a distância do origem ao destino. Se o comprimento do caminho for desconhecido, o contador pode ser inicializado com o valor do pior caso, ou seja, o diâmetro da rede.

Mesmo com um contador de saltos, a inundação ainda pode produzir um número exponencial de pacotes duplicados. Um método mais eficaz para controlar a inundação é rastrear quais pacotes foram enviados para evitar sua retransmissão. Isso pode ser feito inserindo um número de sequência único no cabeçalho de cada pacote. Os roteadores então mantêm uma lista por roteador de origem, registrando quais números de sequência foram vistos. Se um pacote com um número de sequência já presente na lista for recebido, ele não será encaminhado.

Para evitar que essas listas cresçam indefinidamente, elas são gerenciadas com um contador �*k*, indicando que todos os números de sequência até �*k* foram vistos. Verificar se um pacote é uma duplicata torna-se simples ao comparar seu número de sequência com �*k*, descartando-o se for uma duplicata.

A inundação, embora impraticável para a maioria das transmissões de pacotes, tem usos específicos. Ela garante que um pacote alcance todos os nós, útil para a disseminação de informações. Em redes sem fio, todas as mensagens transmitidas por uma estação podem ser recebidas por todas as outras estações, efetivamente implementando a inundação. Alguns algoritmos aproveitam essa propriedade.

A inundação também é robusta, encontrando um caminho mesmo em redes altamente perturbadas, como redes militares em zonas de guerra. Requer uma configuração mínima, com os roteadores precisando apenas conhecer seus vizinhos. A inundação pode servir como um bloco de construção para algoritmos de roteamento mais eficientes e como referência para comparação com outros métodos de roteamento. A inundação sempre escolhe o caminho mais curto, pois todos os caminhos possíveis são selecionados em paralelo, resultando no menor atraso, ignorando o overhead causado pelo próprio processo de inundação.

**5.2.4 Roteamento por Vetor de Distância**

Redes de computadores modernas frequentemente usam algoritmos de roteamento dinâmicos mais eficientes do que a inundação, como o Roteamento por Vetor de Distância (DVR) e o Roteamento de Estado de Enlace (LSR). O Roteamento por Vetor de Distância opera tendo cada roteador manter uma tabela (ou vetor) que fornece a melhor distância conhecida para cada destino e o enlace a ser usado para alcançá-lo. Essas tabelas são atualizadas por meio da troca de informações com os vizinhos, permitindo que cada roteador determine eventualmente a melhor rota para cada destino.

O algoritmo de Roteamento por Vetor de Distância, também conhecido como algoritmo de Bellman-Ford, foi o algoritmo de roteamento original da ARPANET e é usado na Internet sob o nome RIP (Protocolo de Informações de Roteamento).

No DVR, cada roteador mantém uma tabela de roteamento indexada para cada roteador na rede, com cada entrada contendo a interface de saída preferida e uma estimativa da distância até o destino. A métrica usada pode ser o número de saltos ou outra medida, como o atraso de propagação, que os roteadores podem medir diretamente usando pacotes especiais ECHO/REPLY.

Por exemplo, se o atraso for usado como métrica, cada roteador envia periodicamente uma lista de seus atrasos estimados para cada destino para seus vizinhos e recebe listas semelhantes deles. Um roteador então calcula o melhor atraso estimado para alcançar um destino por meio de cada vizinho e atualiza sua tabela de roteamento conforme necessário. Esse processo é ilustrado na figura fornecida, onde o roteador J recebe vetores de atraso de seus vizinhos A, I, H e K e calcula as melhores rotas para vários destinos.

**Problema da Contagem ao Infinito**

Uma grande desvantagem do DVR é o "problema da contagem ao infinito", onde os roteadores podem levar muito tempo para convergir para a rota correta, especialmente quando ocorrem mudanças na rede, como um enlace ou roteador indo para baixo. Boas notícias sobre caminhos mais curtos se propagam rapidamente, mas más notícias sobre caminhos mais longos se espalham lentamente.

Por exemplo, se um roteador inicialmente acredita que tem uma grande distância para um destino e um vizinho relata um caminho mais curto, o roteador rapidamente adota o caminho mais curto. No entanto, se um rota se torna indisponível, o roteador pode continuar a acreditar em um caminho através de um vizinho que também acredita em um caminho através do roteador original, causando uma lenta propagação das más notícias.

Isso é ilustrado na figura fornecida, onde uma rede linear de cinco nós experimenta o problema da contagem ao infinito. Quando o roteador A se torna inativo, outros roteadores gradualmente aumentam suas estimativas de distância para A, eventualmente atingindo um grande número representando o infinito. Este problema é desafiador de resolver, e embora técnicas como reversão envenenada tenham sido propostas, nenhuma foi completamente eficaz na prática.

O capítulo aborda o Roteamento de Estado de Enlace como uma evolução do Roteamento por Vetor de Distância, utilizado na ARPANET até 1979. O problema central com o roteamento por vetor de distância era sua lenta convergência, devido ao problema da contagem ao infinito. Por isso, foi substituído pelo roteamento de estado de enlace, cujas variantes mais comuns são IS-IS e OSPF, amplamente utilizados nas redes atuais.

O roteamento de estado de enlace envolve cinco etapas principais:

1. Descoberta dos vizinhos e aprendizado de seus endereços de rede.
2. Medição da distância ou custo até cada vizinho.
3. Criação de um pacote contendo todas as informações aprendidas.
4. Envio e recebimento desses pacotes entre todos os roteadores.
5. Cálculo dos caminhos mais curtos para cada roteador.

Para determinar os vizinhos, cada roteador envia um pacote "HELLO" em cada linha ponto a ponto. Em redes de broadcast, a situação é mais complexa e pode envolver a modelagem da LAN como um nó único.

O custo de cada enlace é definido automaticamente com base em fatores como largura de banda ou atraso. Os pacotes de estado de enlace contêm informações sobre a identidade do transmissor, número de sequência, TTL e custo para cada vizinho.

A distribuição dos pacotes de estado de enlace é realizada através de flooding, onde cada roteador recebe e envia os pacotes para todos os outros. Problemas como repetição de números de sequência são mitigados com um número de sequência de 32 bits e o uso do TTL.

Após receber todos os pacotes de estado de enlace, cada roteador pode criar um grafo da rede e usar o algoritmo de Dijkstra para calcular os caminhos mais curtos para cada destino possível. Esses caminhos são então inseridos nas tabelas de roteamento.

O roteamento de estado de enlace requer mais memória e cálculos do que o roteamento por vetor de distância, mas é mais eficiente em termos de convergência. Protocolos como IS-IS e OSPF são exemplos comuns de roteamento de estado de enlace utilizados em redes reais.

**5.2.5 Roteamento de Estado de Enlace**

como uma evolução do Roteamento por Vetor de Distância, utilizado na ARPANET até 1979. O problema central com o roteamento por vetor de distância era sua lenta convergência, devido ao problema da contagem ao infinito. Por isso, foi substituído pelo roteamento de estado de enlace, cujas variantes mais comuns são IS-IS e OSPF, amplamente utilizados nas redes atuais.

O roteamento de estado de enlace envolve cinco etapas principais:

1. Descoberta dos vizinhos e aprendizado de seus endereços de rede.
2. Medição da distância ou custo até cada vizinho.
3. Criação de um pacote contendo todas as informações aprendidas.
4. Envio e recebimento desses pacotes entre todos os roteadores.
5. Cálculo dos caminhos mais curtos para cada roteador.

Para determinar os vizinhos, cada roteador envia um pacote "HELLO" em cada linha ponto a ponto. Em redes de broadcast, a situação é mais complexa e pode envolver a modelagem da LAN como um nó único.

O custo de cada enlace é definido automaticamente com base em fatores como largura de banda ou atraso. Os pacotes de estado de enlace contêm informações sobre a identidade do transmissor, número de sequência, TTL e custo para cada vizinho.

A distribuição dos pacotes de estado de enlace é realizada através de flooding, onde cada roteador recebe e envia os pacotes para todos os outros. Problemas como repetição de números de sequência são mitigados com um número de sequência de 32 bits e o uso do TTL.

Após receber todos os pacotes de estado de enlace, cada roteador pode criar um grafo da rede e usar o algoritmo de Dijkstra para calcular os caminhos mais curtos para cada destino possível. Esses caminhos são então inseridos nas tabelas de roteamento.

O roteamento de estado de enlace requer mais memória e cálculos do que o roteamento por vetor de distância, mas é mais eficiente em termos de convergência. Protocolos como IS-IS e OSPF são exemplos comuns de roteamento de estado de enlace utilizados em redes reais.

**5.2.6 O roteamento hierárquico**

surge como uma solução para lidar com o crescimento das redes, que resulta em tabelas de roteamento cada vez maiores e mais complexas. Nesse contexto, os roteadores são divididos em regiões, onde cada roteador conhece os detalhes para rotear pacotes dentro de sua própria região, mas não sabe nada sobre a estrutura interna das outras regiões.

Essa abordagem é semelhante à hierarquia na rede telefônica, onde diferentes áreas são gerenciadas de forma independente. À medida que a rede cresce, podem ser necessários vários níveis de hierarquia, com regiões agrupadas em clusters, clusters em zonas e assim por diante.

Um exemplo de aplicação do roteamento hierárquico é o roteio de um pacote de Berkeley, na Califórnia, para Malindi, no Quênia. Cada nível da hierarquia encaminha o tráfego para o próximo nível, até que o pacote chegue ao seu destino final. Esse método reduz a complexidade das tabelas de roteamento e economiza espaço de armazenamento, embora possa aumentar o comprimento do caminho percorrido pelos pacotes.

Um estudo realizado por Kamoun e Kleinrock em 1979 sugere que o número ideal de níveis hierárquicos para uma rede com N roteadores é o logaritmo natural de N. Isso resultaria em um número total de e ln N entradas por roteador. Eles também demonstraram que o aumento no comprimento médio do caminho causado pelo roteamento hierárquico é geralmente aceitável.

**5.2.7 O roteamento por broadcast**

é utilizado quando um host precisa enviar mensagens para muitos ou todos os outros hosts em uma rede. Existem diferentes métodos para implementar o broadcast, cada um com suas vantagens e desvantagens.

Um método simples é o envio de um pacote específico para cada destino. No entanto, esse método desperdiça largura de banda e exige que a origem tenha uma lista completa de todos os destinos, o que não é prático na maioria dos casos.

Uma melhoria é o roteamento para vários destinos, onde cada pacote contém uma lista de destinos ou um mapa de bits indicando os destinos desejados. Os roteadores determinam as interfaces de saída necessárias e geram uma cópia do pacote para cada uma delas, incluindo apenas os destinos relevantes. Esse método utiliza a largura de banda de forma mais eficiente, mas ainda exige que a origem conheça todos os destinos e pode ser trabalhoso para os roteadores determinarem para onde enviar os pacotes.

O flooding é outra técnica de broadcast, onde cada pacote é enviado por todas as interfaces, exceto pela interface por onde ele chegou. Embora não seja eficiente para comunicação ponto a ponto, o flooding deve ser considerado no broadcast. Porém, quando as rotas de caminho mais curto para os pacotes normais já foram calculadas, o encaminhamento pelo caminho inverso pode ser mais eficiente.

O encaminhamento pelo caminho inverso envia o pacote de broadcast por cada enlace apenas uma vez em cada sentido, o que é eficiente e fácil de implementar. Ele também não requer que os roteadores lembrem dos números de sequência ou listem todos os destinos no pacote.

Um último algoritmo de broadcast faz uso explícito da árvore de escoamento (ou qualquer outra árvore spanning tree) para o roteador que inicia o broadcast. Cada roteador copia o pacote de broadcast para todas as interfaces da spanning tree, exceto para aquela pela qual o pacote chegou. Esse método utiliza a largura de banda de forma mais eficiente, mas requer que cada roteador tenha conhecimento da spanning tree.

**5.2.8 O roteamento por multicast**

É utilizado quando um host precisa enviar mensagens para vários receptores em uma rede. Isso é comum em aplicações como jogos online ou transmissões de vídeo ao vivo para muitos espectadores. Enviar um pacote distinto para cada receptor é ineficiente, assim como o broadcasting, que é um desperdício quando a maioria dos receptores não está interessada na mensagem.

O multicasting envolve o gerenciamento de grupos, onde cada grupo é identificado por um endereço de multicast e os roteadores conhecem os grupos aos quais pertencem. Os esquemas de roteamento por multicast são baseados em esquemas de roteamento por broadcast, mas com o objetivo de enviar pacotes apenas para membros específicos de um grupo, utilizando a largura de banda de forma mais eficiente.

Se o grupo de multicast é denso, o broadcast pode ser um bom ponto de partida, mas os enlaces que não levam a membros do grupo podem ser removidos para criar uma spanning tree eficiente apenas para o grupo em questão. Se o grupo é esparsamente distribuído, uma abordagem baseada em núcleo pode ser mais eficaz, onde uma única spanning tree é calculada para o grupo e os pacotes são enviados para o núcleo antes de serem encaminhados para os membros do grupo.

Existem diferentes métodos para podar uma spanning tree para multicast. Em redes que usam roteamento de estado de enlace, cada roteador pode construir sua própria spanning tree podada para cada grupo. Em redes que usam roteamento por vetor de distância, um algoritmo de encaminhamento pela nota inversa pode ser usado para podar a árvore recursivamente.

As árvores compartilhadas, como as árvores baseadas em núcleo, podem ser uma economia importante em termos de armazenamento, envio de mensagens e computação. Cada roteador precisa manter apenas uma árvore por grupo, em vez de várias árvores, e os roteadores que não fazem parte da árvore não realizam trabalho para dar suporte ao grupo. Isso torna as abordagens de árvore compartilhada ideais para grupos esparsos na Internet.

**5.2.9 O roteamento por anycast**

É um modelo de entrega em que um pacote é entregue ao membro mais próximo de um grupo. É útil quando um serviço é oferecido por vários nós, e o importante é obter a informação correta, não importando qual nó específico é contatado.

Por exemplo, o anycast é utilizado na Internet como parte do DNS (Sistema de Nomes de Domínio), onde vários servidores DNS podem fornecer informações de nomes de domínio, e o importante é contatar o servidor mais próximo em termos de latência.

Felizmente, não é necessário criar novos esquemas de roteamento para o anycast, pois os protocolos de roteamento existentes, como o roteamento por vetor de distância e o roteamento de estado de enlace, podem produzir rotas de anycast. No roteamento por vetor de distância, por exemplo, os nós escolhem o caminho mais curto até o destino anycast, resultando em nós enviando para a ocorrência mais próxima desse destino.

O roteamento de estado de enlace também pode ser usado para anycast, mas com uma consideração adicional para evitar que os protocolos de roteamento encontrem caminhos aparentemente curtos que passem pelo nó anycast. Isso é importante porque as ocorrências do nó anycast são, na realidade, nós localizados em diferentes partes da rede.

No geral, o roteamento por anycast é uma ferramenta útil para garantir a entrega eficiente de serviços em redes distribuídas, onde a proximidade física do cliente ao servidor é importante.

**5.2.10 Roteamento em redes ad hoc**

também conhecidas como MANETs (Mobile Ad hoc NETworks), é uma área desafiadora devido à sua natureza dinâmica e à constante mudança na topologia da rede. Ao contrário das redes fisicamente conectadas, onde a topologia é estável, em redes ad hoc os nós podem se mover, entrar e sair da rede a qualquer momento, o que torna a descoberta e manutenção de rotas muito mais complexas.

Uma característica fundamental das redes ad hoc é a descoberta de rotas apenas quando necessário. Isso significa que as rotas são estabelecidas sob demanda, economizando recursos e reduzindo o overhead na rede. Um algoritmo popular para roteamento em redes ad hoc é o AODV (Ad hoc On-demand Distance Vector), que é semelhante ao algoritmo de roteamento por vetor de distância, mas adaptado para ambientes móveis com largura de banda limitada e duração de bateria baixa.

A descoberta de rota no AODV ocorre quando um nó deseja enviar um pacote para um destino para o qual não possui uma rota conhecida. Nesse caso, o nó emite um pacote ROUTE REQUEST, que é transmitido por broadcast na rede. Cada nó recebe e retransmite esse pacote até que alcance o destino ou um nó que conheça uma rota para o destino. O destino então envia um pacote ROUTE REPLY de volta ao nó remetente, indicando a rota para alcançá-lo.

Para manter as rotas atualizadas devido à mobilidade dos nós, os nós na rede enviam mensagens periódicas chamadas "Hello" para seus vizinhos. Se um nó deixa de receber respostas de um vizinho, ele remove as rotas que passam por esse vizinho de sua tabela de roteamento. Esse processo de manutenção de rotas garante que as rotas permaneçam atualizadas e válidas, mesmo com a mobilidade dos nós.

Além do AODV, existem outros esquemas de roteamento ad hoc, como o DSR (Dynamic Source Routing) e o GPSR (Greedy Perimeter Stateless Routing), cada um com suas próprias características e vantagens. O sucesso desses protocolos dependerá das necessidades específicas das redes ad hoc e das condições de operação.

**5.3 Algoritmos de congestionamento**

O congestionamento em uma rede ocorre quando há mais pacotes presentes do que a rede pode efetivamente transportar, resultando em atrasos e perda de pacotes que afetam o desempenho. Tanto a camada de rede quanto a camada de transporte são responsáveis por lidar com o congestionamento, mas a abordagem mais eficaz envolve reduzir a carga que a camada de transporte coloca sobre a rede.

O congestionamento pode ser ilustrado pelo aumento da carga oferecida à rede, que, quando se aproxima ou excede a capacidade da rede, resulta em perda de pacotes e queda no desempenho. Isso é representado pelo início do congestionamento na figura, onde a quantidade de pacotes entregues pela rede não acompanha o aumento na carga oferecida.

Projetar redes para evitar congestionamento é desejável, mas nem sempre possível. Enlaces ou roteadores com largura de banda limitada podem se tornar gargalos, e a adição de mais memória nem sempre resolve o problema, podendo até piorá-lo em alguns casos. Quando toda a rede está congestionada, a única opção pode ser livrar-se da carga excessiva ou atualizar para uma rede mais rápida.

É importante distinguir entre controle de congestionamento e controle de fluxo. O controle de congestionamento trata da capacidade global da rede em transportar o tráfego oferecido, enquanto o controle de fluxo está relacionado à taxa de transmissão entre um transmissor e um receptor específicos. Embora possam estar interligados, eles abordam problemas diferentes.

O controle de congestionamento pode envolver estratégias para evitar congestionamentos, como gerenciamento de buffer e políticas de descarte de pacotes, bem como técnicas para lidar com o congestionamento quando ele ocorre, como redução de taxa de transmissão e controle de admissão de fluxo. Nosso estudo do controle de congestionamento começará examinando essas técnicas em diferentes escalas de tempo, abordando estratégias para evitar e mitigar congestionamentos.

5.3.1 Controle de técnica de congestionamento

Existem várias técnicas para lidar com o congestionamento em redes, e elas geralmente operam em diferentes escalas de tempo para prevenir ou reagir ao congestionamento, conforme ilustrado na Figura 5.20.

1. **Provisionamento da rede**: Essa técnica envolve aumentar os recursos da rede, como largura de banda e capacidade de roteamento, para acomodar o tráfego. O provisionamento geralmente ocorre em uma escala de tempo de meses, com base em tendências de tráfego a longo prazo.
2. **Roteamento com conhecimento do tráfego**: As rotas são ajustadas de acordo com os padrões de tráfego, redistribuindo o tráfego para evitar congestionamentos. Isso pode envolver a alteração dos pesos dos caminhos mais curtos ou o uso de várias rotas para dividir o tráfego.
3. **Controle de acesso**: Em uma rede de circuito virtual, novas conexões podem ser recusadas se a rede estiver congestionada, evitando que o congestionamento piore.
4. **Controle de tráfego**: Quando o congestionamento é iminente, a rede pode oferecer feedback às fontes responsáveis pelo tráfego, solicitando que elas controlem seu fluxo ou atrasando o tráfego.
5. **Corte de carga**: Quando todas as outras medidas falham, a rede é forçada a descartar pacotes que não podem ser entregues. Uma política eficaz para escolher quais pacotes descartar pode ajudar a evitar o colapso do congestionamento.

A eficácia dessas técnicas depende da capacidade da rede de detectar e responder ao congestionamento de forma rápida e precisa, garantindo que o desempenho da rede seja mantido em níveis aceitáveis mesmo sob condições de tráfego intenso.

**5.3.4 O roteamento com conhecimento do tráfego**

é uma técnica que ajusta as rotas de acordo com o padrão de tráfego da rede. Ao contrário dos esquemas de roteamento que usam pesos de enlace fixos, essa técnica leva em consideração não apenas a topologia da rede, mas também a carga em cada enlace. Isso é feito ajustando os pesos dos enlaces com base na largura de banda, atraso de propagação e carga medida ou atraso médio do enfileiramento.

No entanto, há um risco associado a esse método. Se não for implementado com cuidado, pode levar a oscilações no roteamento, como no exemplo da rede dividida em duas partes Leste e Oeste, conectadas por dois enlaces. Se a carga não for devidamente considerada e apenas a largura de banda e o atraso de propagação forem levados em conta, o problema pode ser evitado, mas em alguns casos isso pode não ser viável.

Duas técnicas podem ajudar a mitigar esse problema: roteamento por caminhos múltiplos e ajuste lento das rotas. O roteamento por caminhos múltiplos permite distribuir o tráfego por várias rotas, enquanto o ajuste lento das rotas permite que o sistema de roteamento se adapte gradualmente às mudanças de carga, evitando oscilações abruptas.

No contexto da Internet, os protocolos de roteamento geralmente não ajustam suas rotas com base na carga, em vez disso, os ajustes são feitos externamente ao protocolo de roteamento, em um processo chamado de engenharia de tráfego. Isso permite que os operadores de rede ajustem manualmente as rotas conforme necessário, sem perturbar o funcionamento básico dos protocolos de roteamento.

**5.3.5 O controle de acesso**

É uma técnica amplamente utilizada em redes de circuito virtual para evitar o congestionamento. A ideia é simples: não estabelecer um novo circuito virtual a menos que a rede possa acomodar o tráfego adicional sem se tornar congestionada. Isso é feito para evitar que mais tráfego seja injetado na rede quando ela já está sobrecarregada, o que só pioraria a situação.

O desafio com essa técnica é determinar quando a adição de um novo circuito virtual resultará em congestionamento. Na rede telefônica, onde a largura de banda das chamadas é fixa, isso é relativamente simples. No entanto, em redes de computadores, onde os circuitos virtuais têm diferentes formas e tamanhos, é necessário caracterizar o tráfego de maneira significativa. Uma abordagem comum é descrever o tráfego em termos de sua velocidade média e forma, usando descritores como token bucket ou leaky bucket, que capturam o efeito das rajadas de tráfego.

Com base nessas descrições de tráfego, a rede pode decidir se admitirá ou não o novo circuito virtual. Isso pode ser feito reservando capacidade suficiente ao longo dos caminhos de cada circuito virtual para evitar o congestionamento. Outra abordagem é usar medições de comportamento passado para estimar quantos circuitos podem ser admitidos sem risco de congestionamento.

O controle de acesso também pode ser combinado com outras técnicas, como o roteamento com conhecimento do tráfego, para estabelecer conexões evitando pontos críticos de congestionamento. Essa abordagem pode envolver o redesenho da rede para evitar roteadores congestionados e suas interfaces.

**5.3.6 O controle de tráfego**

é essencial para manter o bom funcionamento das redes de computadores, especialmente para evitar o congestionamento. Vamos dar uma olhada em algumas técnicas utilizadas para controlar o tráfego em redes de datagramas e circuitos virtuais:

**Monitoramento de Congestionamento:**

Para determinar quando o congestionamento está se aproximando, os roteadores precisam monitorar continuamente os recursos que estão utilizando. Uma forma útil de fazer isso é monitorar o atraso de enfileiramento dentro dos roteadores. Quando esse atraso excede um certo limite, é um sinal de que o congestionamento está começando a ocorrer.

**Feedback aos Transmissores:**

Uma vez que o congestionamento está prestes a ocorrer ou está ocorrendo, os roteadores precisam enviar feedback aos transmissores para reduzir suas transmissões. Existem várias maneiras de fazer isso:

1. **Pacotes Reguladores:**
   * O roteador seleciona um pacote congestionado e envia um pacote regulador de volta ao host de origem, solicitando-lhe que reduza suas transmissões.
   * Para evitar sobrecarregar ainda mais a rede, os roteadores podem enviar pacotes reguladores em um ritmo lento durante momentos de congestionamento.
2. **Notificação Explícita de Congestionamento (ECN):**
   * Os roteadores marcam pacotes que estão passando por congestionamento, definindo um bit no cabeçalho do pacote.
   * O destino observa essa marca e informa ao transmissor quando enviar um pacote de resposta, indicando que o congestionamento está ocorrendo.
   * Isso permite uma redução mais rápida das transmissões, pois o feedback é enviado de volta ao transmissor sem a necessidade de pacotes reguladores adicionais.
3. **Pacotes Reguladores Hop a Hop:**
   * Os pacotes reguladores podem ser enviados a cada hop do caminho do pacote, reduzindo o congestionamento mais rapidamente em comparação com a abordagem tradicional.
   * Embora isso aumente o consumo de buffers nos nós intermediários, oferece alívio imediato no ponto de congestionamento.

Essas técnicas garantem que o tráfego na rede seja controlado de forma eficaz, minimizando o congestionamento e mantendo um desempenho ideal.

**5.3.6 O corte de carga**

é uma estratégia extrema utilizada pelos roteadores para lidar com o congestionamento nas redes de computadores. Assim como as concessionárias de energia elétrica cortam o fornecimento de eletricidade em áreas específicas para evitar o colapso de toda a rede, os roteadores descartam pacotes que não conseguem processar para evitar que a rede entre em colapso devido ao congestionamento.

**Políticas de Descarte de Pacotes:**

1. **Política do Vinho:**
   * Nessa política, pacotes mais antigos são considerados mais importantes que os novos.
   * É preferível manter os pacotes mais antigos para aplicações como transferências de arquivos, onde a ordem dos pacotes é crucial.
2. **Política do Leite:**
   * Pacotes mais novos são considerados mais importantes que os antigos.
   * É preferível manter os pacotes mais recentes para aplicações como mídia em tempo real, onde a entrega pontual dos pacotes é fundamental.

**Colaboração dos Transmissores:**

* Para implementar uma política inteligente de descarte, as aplicações podem marcar seus pacotes para indicar sua importância para a rede.
* Os roteadores podem, então, priorizar o descarte de pacotes menos importantes em situações de congestionamento, preservando os pacotes mais cruciais para a operação da rede.

**Detecção Aleatória Prematura (RED):**

* O RED (Random Early Detection) é um algoritmo que ajuda a reduzir o congestionamento descartando pacotes antes que os buffers dos roteadores fiquem totalmente cheios.
* Os roteadores monitoram o tamanho médio das filas e começam a descartar aleatoriamente uma pequena fração de pacotes quando esse tamanho ultrapassa um certo limite.
* Esse método permite uma ação preventiva contra o congestionamento, notificando implicitamente os transmissores sobre a situação antes que se torne crítica.

Essas estratégias de corte de carga garantem que a rede permaneça operacional mesmo em situações de congestionamento intenso, protegendo-a contra falhas catastróficas.

**5.4 A qualidade de serviço (QoS)**

em redes é crucial para garantir que certas aplicações recebam o desempenho necessário para funcionar corretamente, mesmo em condições de congestionamento. Vamos analisar os quatro aspectos fundamentais abordados para garantir a QoS:

1. Identificação das Aplicações Necessárias:

* É essencial identificar quais aplicações na rede requerem garantias de desempenho mais altas.
* Aplicações de multimídia, por exemplo, frequentemente exigem um throughput mínimo e latência máxima para uma operação satisfatória.

2. Regulação do Tráfego de Entrada:

* Controlar o tráfego que entra na rede é crucial para garantir que as aplicações críticas recebam prioridade.
* Técnicas de controle de acesso e limitação de taxa podem ser empregadas para regular o tráfego e evitar congestionamento.

3. Reserva de Recursos nos Roteadores:

* É necessário reservar recursos nos roteadores para garantir o desempenho das aplicações críticas.
* Mecanismos como a alocação de largura de banda e priorização de pacotes podem ser utilizados para reservar recursos e garantir que as aplicações prioritárias sejam atendidas conforme necessário.

4. Capacidade Adicional com Segurança:

* Determinar se a rede pode aceitar mais tráfego com segurança é crucial para evitar congestionamento e garantir a qualidade de serviço.
* Mecanismos de monitoramento de tráfego e ajuste dinâmico de capacidade podem ser implementados para garantir que a rede possa lidar com aumentos repentinos no tráfego sem comprometer o desempenho das aplicações críticas.

Técnicas de Qualidade de Serviço:

* Não existe uma única técnica que lide eficazmente com todos esses aspectos da qualidade de serviço.
* Diversas técnicas foram desenvolvidas e são combinadas na camada de rede e transporte para garantir a QoS.
* Duas abordagens comuns são os serviços integrados (Integrated Services) e os serviços diferenciados (Differentiated Services), cada uma com suas características e aplicabilidades específicas.

A implementação prática da qualidade de serviço geralmente envolve a combinação de várias dessas técnicas para atender às necessidades específicas das aplicações e garantir um desempenho consistente e confiável da rede.

**5.4 Os requisitos das aplicações** em uma rede podem ser descritos por quatro parâmetros principais: largura de banda, atraso, flutuação e perda. Aqui estão alguns insights sobre as necessidades de várias aplicações comuns:

1. **Correio Eletrônico**:
   * Largura de banda: Baixa
   * Atraso: Baixo
   * Flutuação: Baixa
   * Perda: Média
2. **Transferência de Arquivos**:
   * Largura de banda: Alta
   * Atraso: Baixo
   * Flutuação: Baixa
   * Perda: Média
3. **Acesso à Web**:
   * Largura de banda: Média
   * Atraso: Médio
   * Flutuação: Baixa
   * Perda: Média
4. **Login Remoto**:
   * Largura de banda: Baixa
   * Atraso: Médio
   * Flutuação: Média
   * Perda: Média
5. **Áudio por Demanda**:
   * Largura de banda: Baixa
   * Atraso: Baixo
   * Flutuação: Alta
   * Perda: Baixa
6. **Vídeo por Demanda**:
   * Largura de banda: Alta
   * Atraso: Baixo
   * Flutuação: Alta
   * Perda: Baixa
7. **Telefonia**:
   * Largura de banda: Baixa
   * Atraso: Alto
   * Flutuação: Alta
   * Perda: Baixa
8. **Videoconferência**:
   * Largura de banda: Alta
   * Atraso: Alto
   * Flutuação: Alta
   * Perda: Baixa

Esses requisitos variam dependendo do tipo de aplicação e da sensibilidade de cada uma a fatores como atraso, flutuação e perda de pacotes. Por exemplo, aplicações sensíveis ao tempo, como telefonia e videoconferência, têm requisitos rigorosos de atraso, flutuação e perda, enquanto outras aplicações, como transferência de arquivos, podem tolerar mais atraso e perda. A compreensão desses requisitos é fundamental para o design e a implementação de redes que atendam adequadamente às necessidades das aplicações.

**5.4.1 A modelagem de tráfego**

é essencial para garantir a qualidade de serviço (QoS) em redes, especialmente em ambientes onde o tráfego é bursty (em rajadas). Dois algoritmos comuns usados para essa finalidade são o "leaky bucket" (balde furado) e o "token bucket" (balde de fichas).

1. **Leaky Bucket**:
   * Neste modelo, imagina-se um balde com um pequeno furo no fundo. O tráfego que entra no balde representa os pacotes que chegam à rede.
   * Independentemente da taxa de entrada de tráfego, a saída ocorre a uma taxa constante quando há água no balde.
   * Quando o balde está cheio (capacidade máxima), qualquer tráfego adicional que chega é descartado.
   * Esse modelo é utilizado para controlar o tráfego que entra na rede, garantindo que não exceda uma determinada taxa média.
   * Pacotes que chegam quando o balde está cheio podem ser descartados ou terem sua prioridade reduzida.
2. **Token Bucket**:
   * Neste modelo, imagina-se um balde sendo enchido com "fichas" (tokens) a uma taxa constante.
   * Cada pacote que deseja ser transmitido pela rede precisa "pegar" um token do balde.
   * O balde tem uma capacidade máxima de tokens que podem ser acumulados.
   * Se não houver tokens disponíveis quando um pacote chegar, ele precisa esperar até que tokens suficientes sejam acumulados no balde.
   * Esse modelo também é usado para controlar a taxa de tráfego, permitindo rajadas de curto prazo, desde que não excedam a capacidade do balde.

Ambos os modelos são úteis para limitar a taxa de tráfego em longo prazo, enquanto permitem rajadas de curto prazo até um certo limite. Isso ajuda a suavizar o tráfego na rede, reduzindo o congestionamento e garantindo que os requisitos de QoS sejam atendidos.

Os algoritmos de leaky e token bucket são fáceis de implementar e oferecem maneiras eficazes de controlar o tráfego na rede, garantindo que ele esteja dentro dos limites especificados pelos acordos de nível de serviço (SLAs) entre os clientes e os provedores de rede.

**5.4.2 Listagem de pacotes**

A capacidade de regular a forma do tráfego oferecido é um bom ponto de partida para garantir a qualidade de serviço (QoS) em uma rede. No entanto, para oferecer uma garantia de desempenho, é necessário reservar recursos suficientes ao longo da rota que os pacotes percorrem na rede. Isso é especialmente importante quando se considera que todos os pacotes de um fluxo seguem a mesma rota, pois dispersar os pacotes pelos roteadores ao acaso dificulta estabelecer qualquer garantia de QoS.

Os algoritmos que alocam recursos do roteador entre os pacotes de um fluxo e entre fluxos concorrentes são chamados algoritmos de escalonamento de pacotes. Esses algoritmos podem reservar três tipos de recursos para diferentes fluxos: largura de banda, espaço em buffer e ciclos de CPU.

A largura de banda é um recurso óbvio a ser reservado, garantindo que a capacidade de transmissão de uma interface de saída não seja sobrecarregada pelos fluxos que a utilizam. O espaço em buffer é essencial para absorver pequenas rajadas de tráfego e evitar a perda de pacotes devido à falta de espaço de armazenamento temporário nos roteadores. Por fim, os ciclos de CPU são necessários para processar os pacotes, e garantir que a CPU não seja sobrecarregada é crucial para garantir um processamento oportuno e eficiente dos pacotes.

Os algoritmos de escalonamento de pacotes determinam quais pacotes no buffer serão enviados na interface de saída seguinte, alocando eficientemente os recursos do roteador. Um dos algoritmos mais simples é o FIFO (First-In, First-Out), que envia os pacotes na mesma ordem em que chegaram. No entanto, o FIFO não é adequado para garantir uma boa QoS em redes com múltiplos fluxos, pois um fluxo agressivo pode monopolizar os recursos dos roteadores, prejudicando outros fluxos.

Para lidar com esse problema, foram desenvolvidos algoritmos mais sofisticados, como o enfileiramento ordenado com rodízio de filas, que divide os recursos de forma mais equitativa entre os diferentes fluxos. Outro exemplo é o enfileiramento ordenado com rodízio de filas ponderado (WFQ), que atribui pesos aos fluxos para garantir uma distribuição justa dos recursos. Esses algoritmos oferecem melhor isolamento entre os fluxos e evitam a interferência entre eles.

Outras abordagens incluem o escalonamento por prioridade, onde os pacotes são classificados com base em sua prioridade e os de alta prioridade são enviados antes dos de baixa prioridade, e o escalonamento baseado em registros de tempo, onde os pacotes são enviados na ordem de seus registros de tempo para garantir um atraso mais consistente na entrega dos pacotes pela rede.

Em resumo, os algoritmos de escalonamento de pacotes desempenham um papel crucial na garantia de QoS em redes de computadores, garantindo uma distribuição eficiente e equitativa dos recursos do roteador entre os diferentes fluxos de tráfego.

**5.4.4 O controle de acesso**

É essencial para garantir a qualidade de serviço (QoS) em redes de computadores. Agora que entendemos como modelar o tráfego e reservar recursos ao longo da rota, é hora de colocar tudo isso em prática para oferecer a QoS desejada. O processo de controle de acesso é fundamental para estabelecer essas garantias.

O controle de acesso tem um papel crucial na QoS. Primeiro, é utilizado para regular o congestionamento, proporcionando uma garantia de desempenho, embora seja fraca. As garantias de QoS que estamos considerando agora são mais robustas, mas seguem o mesmo modelo. O usuário oferece um fluxo à rede com um requisito de QoS desejado, e a rede decide se deve admitir ou rejeitar esse fluxo com base em sua capacidade e nos compromissos assumidos com outros fluxos.

As reservas de recursos devem ser feitas em todos os roteadores ao longo da rota que os pacotes percorrerão. A falta de reservas em qualquer roteador ao longo do caminho pode levar a congestionamentos, e um único roteador congestionado pode comprometer a garantia de QoS. Algoritmos de roteamento por QoS são usados para encontrar o melhor caminho para os fluxos, levando em consideração a capacidade de reserva disponível.

Além disso, é essencial descrever com precisão os fluxos em termos de parâmetros específicos, conhecidos como especificação de fluxo. Isso permite que os roteadores avaliem se podem atender aos requisitos de QoS de um determinado fluxo e, se necessário, façam ajustes ao longo do caminho.

Um método comum para relacionar especificações de fluxo aos recursos do roteador é baseado em token buckets e Weighted Fair Queueing (WFQ). Cada fluxo recebe um peso de WFQ grande o suficiente para drenar sua taxa de token bucket, garantindo assim uma largura de banda mínima e limitando as rajadas de tráfego. Isso garante que cada fluxo atenda às suas garantias de largura de banda e atraso, independentemente do comportamento dos outros fluxos.

Em resumo, o controle de acesso desempenha um papel fundamental na garantia da qualidade de serviço em redes de computadores, permitindo que a rede tome decisões inteligentes sobre a aceitação e o encaminhamento de fluxos com base em seus requisitos de QoS.

**5.4.5 Os Serviços Integrados**

desenvolvidos pela IETF entre 1995 e 1997, foram uma tentativa significativa de criar uma arquitetura para streaming de multimídia, cobrindo tanto aplicações unicast quanto multicast. Aqui, focaremos no multicast, onde um transmissor envia dados para múltiplos receptores.

O Protocolo de Reserva de Recursos (RSVP) é uma parte crucial da arquitetura de Serviços Integrados. Ele permite que transmissores enviem dados para múltiplos grupos de receptores, permitindo que receptores individuais mudem de canais livremente e otimizando o uso da largura de banda enquanto eliminam o congestionamento.

Em sua forma mais simples, o RSVP utiliza roteamento por multicast com árvores de abrangência (spanning trees). Cada grupo de receptores recebe um endereço, e os transmissores colocam esse endereço em seus pacotes de dados. Um algoritmo de roteamento por multicast padrão é então utilizado para construir uma árvore de abrangência que conecta todos os membros do grupo.

Quando um receptor deseja receber dados de um determinado grupo, ele envia uma mensagem de reserva pela árvore até o transmissor, utilizando o algoritmo de encaminhamento pelo caminho inverso. Em cada hop, os roteadores reservam a largura de banda necessária para o fluxo de dados. Se a largura de banda disponível não for suficiente, o RSVP informa a falha. Uma vez que a mensagem de reserva retorna ao transmissor, a largura de banda já está reservada ao longo de todo o caminho entre o transmissor e o receptor.

Para aplicações em que os grupos de receptores podem mudar dinamicamente, o RSVP permite que os receptores especifiquem uma ou mais origens das quais desejam receber informações. Isso permite otimizar o planejamento da largura de banda, garantindo que dois receptores só compartilhem um caminho se ambos concordarem em não alterar as origens posteriormente.

Em resumo, o RSVP desempenha um papel crucial na arquitetura de Serviços Integrados, permitindo a reserva dinâmica de largura de banda para transmissões multicast, garantindo uma distribuição eficiente de dados e evitando congestionamentos.

**5.4.6 Os Serviços Diferenciados (DS)**

são uma abordagem mais simples para oferecer qualidade de serviço em comparação com os Serviços Integrados, pois não exigem configuração antecipada, reserva de recursos ou negociação ponto a ponto para cada fluxo. Essa estratégia, também conhecida como qualidade de serviço baseada em classe, foi padronizada pela IETF na arquitetura de Serviços Diferenciados, descrita nas RFCs 2474, 2475 e várias outras.

Nesse modelo, os pacotes são marcados com a classe a que pertencem, o que é executado no campo de serviços diferenciados dos pacotes IPv4 e IPv6. Essas classes representam comportamentos por hop, ou seja, o tratamento que cada pacote receberá em cada roteador ao longo de sua rota, não garantindo uma qualidade de serviço pela rede. Alguns pacotes podem receber um serviço melhor, enquanto outros recebem um serviço regular. O tráfego dentro de uma classe pode precisar obedecer a algumas regras específicas, como um controle de fluxo com uma taxa de escoamento especificada.

As operadoras podem definir diferentes classes de serviço e regras de encaminhamento correspondentes. Por exemplo, uma classe de serviço pode garantir baixo atraso e baixa probabilidade de descarte de pacotes, enquanto outra classe pode oferecer um serviço regular. Essa abordagem é flexível e pode ser adaptada de acordo com as necessidades e políticas da operadora.

Para tornar mais clara a diferença entre qualidade de serviço baseada em fluxo e qualidade de serviço baseada em classe, o exemplo da telefonia na Internet é frequentemente citado. Com um esquema baseado em fluxo, cada chamada telefônica recebe recursos e garantias próprios, enquanto com um esquema baseado em classe, todas as chamadas telefônicas compartilham os recursos reservados para a classe de telefonia, sem recursos privados reservados apenas para cada chamada.

A implementação das classes de serviço pode variar, mas uma abordagem comum é classificar os pacotes em diferentes classes de prioridade e tratá-los de acordo com essas prioridades em cada roteador ao longo do caminho. Por exemplo, pode-se usar um escalonador de pacotes que dê prioridade às classes de serviço mais altas, garantindo que elas recebam mais largura de banda e menor atraso.

Em resumo, os Serviços Diferenciados oferecem uma maneira mais simples e flexível de fornecer qualidade de serviço em comparação com os Serviços Integrados, permitindo que operadoras de rede definam classes de serviço e regras de encaminhamento de acordo com suas necessidades e políticas.

**5.5 Interligar redes**

é uma tarefa complexa devido à diversidade de tecnologias e protocolos usados em cada camada das redes individuais. Desde PANs (Redes de Área Pessoal) até WANs (Redes de Área Ampla), cada tipo de rede pode utilizar diferentes tecnologias e protocolos, como Ethernet, redes de telefone fixo e móvel, 802.11 e 802.16, entre outros. Quando essas redes são interconectadas para formar uma rede interligada, ou "internet" com "i" minúsculo, surgem várias questões que precisam ser abordadas.

Seria muito mais simples interconectar as redes se todas usassem a mesma tecnologia, mas na realidade, isso não é o caso. A multiplicidade de tecnologias é uma condição que provavelmente não desaparecerá tão cedo, pois diferentes tipos de redes resolvem diferentes problemas e atendem a diferentes necessidades. Mesmo que haja um tipo dominante de rede, como a Ethernet, a heterogeneidade das redes veio para ficar.

Apesar das dificuldades decorrentes da heterogeneidade das redes, há um incentivo para interconectá-las. Bob Metcalfe observou que o valor de uma rede com N nós é proporcional ao quadrado do número de nós (N²). Isso significa que grandes redes, que permitem mais conexões entre os nós, são muito mais valiosas do que redes pequenas. Portanto, sempre haverá um incentivo para combinar redes menores.

A Internet é o principal exemplo de interconexão de redes. Ao juntar diferentes redes, a Internet permite que usuários em qualquer uma delas se comuniquem com usuários em todas as outras. Quando você paga por um serviço de Internet, está pagando pela capacidade de trocar pacotes com qualquer outro host conectado à Internet, não apenas com hosts na mesma rede local.

Interligar redes requer resolver problemas de heterogeneidade e de escala. As redes interligadas podem diferir em vários aspectos, como tecnologia de transmissão, endereçamento, protocolos de roteamento, etc. Para lidar com esses desafios, protocolos como o IP (Internet Protocol) utilizam técnicas como tunelamento em redes, roteamento em redes interligadas e fragmentação de pacotes. Essas técnicas permitem que os pacotes sejam encaminhados de forma eficiente através de diferentes redes, garantindo a conectividade entre diferentes sistemas e usuários em toda a internet.

**5.5.1 Diferencia entre as redes**

As redes podem diferir em vários aspectos, especialmente na camada de rede, o que torna a interligação de redes uma tarefa desafiadora. Algumas dessas diferenças incluem:

1. **Endereçamento:** Uma origem em uma rede deve ser capaz de endereçar o destino, o que pode ser complicado quando as redes de origem e destino utilizam diferentes tecnologias ou formatos de endereçamento.
2. **Multicast:** Como realizar o multicast de pacotes para um grupo com membros em uma rede que não suporta multicast pode ser um desafio.
3. **Tamanho máximo de pacotes:** Redes distintas podem ter diferentes tamanhos máximos de pacotes. Passar um pacote grande por uma rede com um tamanho máximo de pacote menor requer fragmentação e remontagem do pacote.
4. **Orientação de conexão:** Quando os pacotes de uma rede orientada a conexões transitam por uma rede não orientada a conexões, eles podem chegar fora de ordem, causando problemas para o transmissor e o receptor.
5. **Qualidade de Serviço (QoS):** Redes podem oferecer diferentes níveis de QoS, o que torna difícil garantir largura de banda e atraso para tráfego em tempo real de ponta a ponta.
6. **Segurança:** As redes podem ter diferentes mecanismos de segurança, o que pode complicar a garantia de confidencialidade e integridade dos dados quando trafegam entre redes interligadas.
7. **Contabilidade:** Diferenças na contabilidade, como tarifas de roaming em redes de celular, podem levar a contas inesperadamente altas para os usuários quando o uso normal se torna caro.

Para superar essas diferenças, os gateways entre redes podem realizar funções como gerar pacotes separados para cada destino, fragmentar e remontar pacotes grandes, bufferizar pacotes para entrega em ordem e aplicar criptografia para confidencialidade e integridade dos dados. No entanto, algumas diferenças, como na qualidade de serviço, podem ser mais difíceis de reconciliar e podem exigir compromissos ou adaptações adicionais.

**5.5.2 como as redes podem ser conectadas**

Existem duas abordagens principais para conectar redes diferentes:

1. **Tradução ou Conversão de Pacotes:** Nesta abordagem, dispositivos são colocados nas fronteiras entre as redes para traduzir ou converter pacotes de um tipo de rede para outro. Isso envolve processamento adicional nos limites entre as redes para garantir que os pacotes sejam compatíveis com as tecnologias e formatos de cada rede. Essa abordagem é útil quando as redes são muito diferentes e não compartilham uma camada de abstração comum.
2. **Camada de Abstração Comum:** Essa abordagem envolve a criação de uma camada comum em cima das diferentes redes para ocultar as diferenças entre elas. Dispositivos são colocados nas fronteiras entre as redes para garantir a interconexão. A ideia é criar uma camada de abstração que permita que as redes diferentes se comuniquem de forma transparente, sem a necessidade de tradução ou conversão de pacotes. Esta abordagem tem sido tremendamente bem-sucedida e é exemplificada pelo protocolo IP (Internet Protocol).

Os roteadores são dispositivos-chave na interconexão de redes, pois operam na camada de rede e são responsáveis por encaminhar pacotes entre diferentes redes. Eles são capazes de examinar o endereço de destino de um pacote e determinar para qual rede deve ser enviado. Os switches (ou bridges) também são usados para conectar redes na camada de enlace, mas geralmente são usados para conectar o mesmo tipo de rede, enquanto os roteadores lidam com diferentes tipos de redes na camada de rede.

A interconexão de redes tem sido bem-sucedida na montagem de grandes redes, especialmente quando há uma camada de rede comum, como o IP, que oferece um formato de pacote universal reconhecido por quase todas as redes. No entanto, surgiram muitos protocolos de rede ao longo do tempo, o que torna difícil garantir que todas as redes possam se comunicar perfeitamente. Alguns exemplos incluem o IPX, SNA e AppleTalk, embora o IPv4 e o IPv6 sejam os mais relevantes atualmente.

Roteadores multiprotocolo são capazes de lidar com vários protocolos de rede, mas traduzir pacotes entre redes diferentes pode ser difícil e propenso a falhas, especialmente quando os protocolos são fundamentalmente diferentes. A interoperabilidade entre IPv4 e IPv6, por exemplo, tem sido um desafio significativo. No entanto, o IP tem funcionado como um denominador comum suficiente para permitir a interconexão de redes heterogêneas, embora com algumas limitações em termos de serviços oferecidos.

**5.5.3 Tunelamento**

O tunelamento é uma técnica comum e eficaz para lidar com a interligação de duas redes, especialmente quando os hosts de origem e de destino estão no mesmo tipo de rede, mas há uma rede de outro tipo entre eles. Esta técnica é frequentemente usada em cenários como o de um banco internacional com escritórios em diferentes países, onde cada escritório pode estar em uma rede IPv6 local, mas a conectividade entre eles é fornecida pela Internet IPv4.

O processo de tunelamento envolve encapsular pacotes de uma rede dentro de pacotes de outra rede. Por exemplo, para enviar um pacote IPv6 de Paris a Londres através da Internet IPv4, um host em Paris encapsula o pacote IPv6 dentro de um pacote IPv4 e o envia para o roteador multiprotocolo que conecta a rede IPv6 de Paris à Internet IPv4. Esse roteador, então, encaminha o pacote através da Internet IPv4 para o roteador multiprotocolo em Londres, que extrai o pacote IPv6 original e o entrega ao host de destino em Londres.

O tunelamento é semelhante a transportar um carro de um país para outro através de um túnel subaquático, como o Eurotúnel. Na analogia, o carro representa o pacote, e o túnel é a conexão entre as redes. O tunelamento é usado para conectar hosts e redes isoladas usando outras redes, criando uma rede overlay. No entanto, uma limitação do tunelamento é que os hosts dentro do túnel não podem ser alcançados diretamente, pois os pacotes não podem escapar do túnel. No entanto, essa limitação é transformada em uma vantagem com as VPNs (Virtual Private Networks), que são overlays usados para fornecer segurança.

Além disso, ao lidar com o roteamento entre redes interligadas, surgem várias complicações, como diferentes algoritmos de roteamento, políticas comerciais entre ISPs e questões legais, especialmente quando o roteamento envolve a travessia de fronteiras internacionais. O Border Gateway Protocol (BGP) é comumente usado para roteamento interdomínio na Internet, e as políticas de roteamento desempenham um papel importante na seleção de rotas pelas redes autônomas.

**5.5.4 O roteamento entre redes interligadas**

apresenta desafios adicionais em comparação com o roteamento dentro de uma única rede. Existem várias complicações a serem consideradas:

1. **Diferentes algoritmos de roteamento:** As redes podem usar algoritmos de roteamento internamente diferentes, como roteamento de estado de enlace e roteamento por vetor de distância. Isso pode dificultar a determinação dos caminhos mais curtos na rede interligada.
2. **Diferenças nas políticas dos operadores:** Operadores de rede podem ter objetivos diferentes ao determinar as rotas. Por exemplo, um operador pode priorizar o menor atraso, enquanto outro pode priorizar o custo mais baixo. Isso pode levar a diferentes definições de custo para os caminhos mais curtos, tornando-os menos bem definidos na rede interligada.
3. **Confidencialidade das informações:** Operadores podem querer manter informações sobre os detalhes das rotas em suas redes confidenciais, especialmente se isso envolver informações comerciais sensíveis, como custos monetários.
4. **Hierarquia de redes:** Redes interligadas podem ser significativamente maiores do que qualquer uma das redes individuais que as compõem. Isso pode exigir algoritmos de roteamento que se adaptem bem a uma hierarquia, mesmo que as redes individuais não precisem usar uma.

Para lidar com essas complicações, geralmente são usados dois níveis de algoritmos de roteamento:

* **Protocolos de gateway interior (intradomínio):** Usados dentro de cada rede individual para roteamento interno. Podem ser diferentes em redes diferentes.
* **Protocolos de gateway exterior (interdomínio):** Usados entre as redes que compõem a rede interligada. Todas as redes devem usar o mesmo protocolo para roteamento entre domínios. Na Internet, o protocolo de roteamento interdomínio é o BGP (Border Gateway Protocol).

Cada rede individual é normalmente chamada de Sistema Autônomo (AS) e é operada independentemente das outras. Os ASs podem ser representados por redes de ISP, e cada ISP pode ser composto de mais de um AS, dependendo da forma como é gerenciado ou adquirido.

Os protocolos de roteamento interdomínio, como o BGP, são influenciados por fatores comerciais, acordos entre ISPs e questões legais, como leis de privacidade. As políticas de roteamento desempenham um papel importante na seleção de rotas pelas redes autônomas, e os arranjos comerciais entre ISPs podem afetar significativamente o roteamento entre redes interligadas.

**5.5.5 A fragmentação de pacotes**

é necessária quando um pacote é maior do que o tamanho máximo permitido em uma determinada rede ou enlace. Existem duas estratégias principais para lidar com a fragmentação de pacotes:

1. **Fragmentação transparente:** Nesta estratégia, os roteadores intermediários são responsáveis por fragmentar o pacote em pedaços menores e recombina-los no destino final. Isso permite que os fragmentos passem por redes com diferentes tamanhos máximos de pacote sem que as redes subsequentes estejam cientes da fragmentação. No entanto, esta abordagem pode introduzir overhead adicional, exigindo que os roteadores façam mais trabalho para manter e reconstruir os fragmentos.
2. **Fragmentação não transparente:** Nesta estratégia, os roteadores simplesmente tratam cada fragmento como se fosse um pacote completo, passando-os através da rede sem tentar recombiná-los. A reconstrução do pacote original é deixada para o host de destino. Embora esta abordagem possa exigir menos trabalho dos roteadores intermediários, pode resultar em maior overhead devido à necessidade de fragmentação e retransmissão de pacotes no destino.

A estratégia moderna preferida na Internet é evitar a fragmentação de pacotes na rede sempre que possível. Em vez disso, a descoberta da MTU do caminho (Path MTU Discovery) é usada. Neste método, cada pacote IP é enviado com um indicador de que não deve ser fragmentado. Se um roteador ao longo do caminho encontrar um pacote muito grande, ele retorna um pacote de erro para a origem indicando a necessidade de fragmentação. A origem então ajusta o tamanho dos pacotes subsequentes com base nessa informação. Embora isso possa introduzir atrasos adicionais devido ao processo de descoberta, evita os problemas associados à fragmentação de pacotes na rede e permite uma melhor adaptação aos requisitos de tamanho de pacote ao longo do caminho.

**5.6 A camada de rede da Internet**

Os princípios que moldaram o projeto da Internet são fundamentais para entender por que ela é tão bem-sucedida. Aqui estão os dez principais princípios, conforme resumidos:

1. **Certifique-se de que funciona:** Protótipos devem ser testados e capazes de se comunicar com sucesso antes que um padrão seja finalizado. Isso evita a criação de padrões defeituosos que exigem revisões constantes.
2. **Mantenha a simplicidade:** Prefira soluções simples sempre que possível. Evite recursos desnecessários e escolha a abordagem mais simples para resolver problemas.
3. **Faça escolhas claras:** Evite opções e modos redundantes. Opte por uma abordagem clara e direta para executar uma determinada ação.
4. **Explore a modularidade:** Divida o projeto em módulos independentes para facilitar a manutenção e permitir alterações em partes específicas sem afetar o sistema como um todo.
5. **Espere heterogeneidade:** Reconheça que diferentes tipos de hardware, instalações de transmissão e aplicações coexistirão em uma grande rede. Projetos devem ser flexíveis o suficiente para lidar com essa diversidade.
6. **Evite opções e parâmetros estáticos:** Se possível, permita que os transmissores e receptores negociem valores dinâmicos em vez de usar opções fixas.
7. **Procure um bom projeto; ele não precisa ser perfeito:** Em vez de tentar resolver todos os casos especiais complicados, concentre-se em um projeto sólido e delegue a resolução dos casos complexos para as partes interessadas.
8. **Seja rígido ao enviar e tolerante ao receber:** Siga rigorosamente os padrões ao enviar pacotes, mas esteja preparado para lidar com pacotes recebidos que possam não estar em conformidade.
9. **Pense na escalabilidade:** Projetos devem ser capazes de lidar efetivamente com um grande número de hosts e usuários. Isso pode exigir uma distribuição uniforme da carga pelos recursos disponíveis.
10. **Considere desempenho e custo:** Redes com desempenho fraco ou custos elevados podem não ser amplamente adotadas. Portanto, é importante equilibrar o desempenho com os custos de implementação e operação.

Esses princípios formam a base do projeto da Internet e contribuíram significativamente para sua robustez e sucesso. Eles garantem que a Internet seja uma infraestrutura flexível e adaptável, capaz de suportar uma ampla variedade de aplicativos e dispositivos.

**5.6.1 O protocolo IP versão 4 (IPv4)**

O cabeçalho do IPv4 é composto por várias partes essenciais, cada uma com sua função específica. Aqui está uma explicação detalhada das principais partes do cabeçalho IPv4:

1. **Versão:** Este campo indica a versão do protocolo IP que está sendo utilizado. Atualmente, é dominado pela versão 4 (IPv4).
2. **IHL (Internet Header Length):** Este campo especifica o tamanho do cabeçalho em palavras de 32 bits. O valor mínimo é 5, indicando um cabeçalho de tamanho fixo de 20 bytes. O valor máximo é 15, o que limita o cabeçalho a 60 bytes, permitindo um espaço máximo de 40 bytes para opções.
3. **Serviços diferenciados:** Originalmente chamado de Tipo de Serviço, este campo é usado para distinguir entre diferentes classes de serviços. Ele fornece informações sobre prioridade, confiabilidade e velocidade.
4. **Identificação:** Este campo é usado para ajudar o host de destino a determinar a qual datagrama pertence um fragmento recém-chegado. Todos os fragmentos de um datagrama têm o mesmo valor de identificação.
5. **DF (Don't Fragment) e MF (More Fragments):** O bit DF indica que o datagrama não deve ser fragmentado, enquanto o bit MF é usado para indicar se há mais fragmentos além do atual.
6. **Deslocamento de Fragmento:** Este campo indica a posição do fragmento atual no datagrama original.
7. **TTL (Time to Live):** Este campo é um contador usado para limitar a vida útil dos pacotes. É decrementado a cada hop e, quando chega a zero, o pacote é descartado.
8. **Protocolo:** Este campo indica a que processo de transporte o datagrama deve ser entregue, como TCP ou UDP.
9. **Checksum do cabeçalho:** Este campo é usado para verificar a integridade do cabeçalho IP durante a transmissão.
10. **Endereço de Origem e Destino:** Esses campos indicam os endereços IP das interfaces de rede de origem e destino.
11. **Opções:** Este campo foi projetado para permitir que versões posteriores do protocolo incluam informações adicionais. Existem várias opções, como segurança, roteamento estrito ou solto, registro de rota e timestamp.

O cabeçalho IPv4 desempenha um papel fundamental no encaminhamento de pacotes pela Internet, fornecendo informações essenciais para garantir que os dados sejam entregues corretamente e de forma eficiente.

Formato dos Endereços IP: Os endereços IPv4 são representados por 32 bits e são usados para identificar hosts e roteadores na Internet. Cada endereço refere-se a uma interface de rede e não diretamente a um host.

Hierarquia e Prefixos: Os endereços IP são hierárquicos e consistem em uma parte de rede e uma parte de host. A parte de rede identifica a rede à qual o host pertence, enquanto a parte de host identifica o host específico nessa rede. Os blocos contíguos de espaço de endereços IP são chamados de prefixos, e cada prefixo é escrito com um número que indica o tamanho da parte de rede.

Notação Decimal com Ponto: Os endereços IP são escritos em notação decimal com ponto, onde cada byte é representado em decimal de 0 a 255. Por exemplo, o endereço IP 128.208.2.151 corresponde ao endereço hexadecimal 80D00297.

Máscara de Sub-Rede: A máscara de sub-rede é uma representação binária dos bits da parte de rede do endereço IP. Ela é usada para extrair a parte de rede do endereço IP por meio de uma operação lógica AND.

Vantagens e Desvantagens da Hierarquia de Endereços: A principal vantagem dos prefixos é a capacidade dos roteadores de encaminhar pacotes com base apenas na parte de rede do endereço, o que simplifica as tabelas de roteamento. No entanto, essa abordagem torna os endereços IP dependentes do local, o que requer soluções como o IP móvel para suportar hosts que se movem entre redes. Além disso, a hierarquia pode resultar em desperdício de endereços se não for gerenciada com cuidado.

Escassez de Endereços IPv4 e a Transição para IPv6: A Internet tem enfrentado uma rápida escassez de endereços IPv4 devido ao crescimento exponencial da rede. O IPv6 foi desenvolvido como uma solução para esse problema, oferecendo um espaço de endereçamento muito maior. No entanto, a transição para IPv6 ainda está em andamento. Enquanto isso, medidas para alocar endereços IPv4 de maneira mais eficiente são necessárias.

A subdivisão de redes em sub-redes é uma estratégia importante para lidar com o crescimento e a segmentação de redes de forma eficiente. Aqui estão os principais pontos discutidos:

1. **Controle de Endereços IP pela ICANN**: A ICANN é responsável pelo controle e atribuição de números de rede para evitar conflitos. Ela delega partes do espaço de endereços a autoridades regionais, que distribuem endereços IP aos provedores de serviços de Internet (ISPs) e outras organizações.
2. **Necessidade de Subdivisão de Redes**: À medida que as empresas crescem, há uma demanda contínua por mais endereços IP. No entanto, a atribuição de endereços IP em blocos fixos pode levar a problemas de utilização eficiente, especialmente se um bloco inicial for muito grande para as necessidades atuais.
3. **Sub-Redes como Solução**: A subdivisão de uma rede em sub-redes permite que uma organização aloque endereços IP internamente de forma mais flexível, mantendo a aparência de uma única rede externamente. Cada sub-rede é tratada como uma rede separada, mas todas estão conectadas internamente.
4. **Divisão de um Bloco de Endereços**: Um bloco inicial de endereços IP pode ser dividido em várias sub-redes, cada uma com seu próprio prefixo e máscara de sub-rede. A divisão pode ser feita de forma desigual para atender às necessidades específicas de cada parte da organização.
5. **Roteamento Baseado em Sub-Redes**: Os roteadores usam máscaras de sub-rede para determinar a qual sub-rede um pacote pertence. Eles realizam uma operação lógica AND entre o endereço de destino do pacote e a máscara de sub-rede de cada sub-rede para determinar a rota correta.
6. **Flexibilidade e Manutenção**: A subdivisão de redes em sub-redes oferece flexibilidade para adaptar a rede às necessidades em constante mudança da organização. As mudanças na subdivisão de redes podem ser feitas internamente, sem a necessidade de intervenção externa.
7. **CIDR (Classless Inter-Domain Routing)**: O CIDR é uma abordagem de endereçamento IP que permite uma divisão mais flexível e eficiente do espaço de endereços, sem depender das classes tradicionais de endereços IP. Ele simplifica o roteamento e a atribuição de endereços IP em redes grandes.

O CIDR, ou Classless Inter-Domain Routing (Roteamento Interdomínio Sem Classes), é uma técnica crucial para lidar com o problema da explosão da tabela de roteamento. Aqui estão os principais pontos sobre o CIDR:

Explosão da Tabela de Roteamento: Com o crescimento da Internet, o número de entradas na tabela de roteamento nos roteadores centrais aumenta exponencialmente. Isso se deve à necessidade de cada roteador conhecer as rotas para todas as redes conectadas à Internet.

Roteamento na Internet: Roteadores em diferentes locais precisam conhecer as rotas para redes específicas. Por exemplo, um roteador em uma universidade precisa conhecer as rotas para as sub-redes locais e também para outras redes na Internet.

Agregação de Rotas: A agregação de rotas é uma técnica na qual vários prefixos pequenos são combinados em um único prefixo maior. Isso reduz o número de entradas na tabela de roteamento, simplificando o processo de roteamento.

Prefixos de Tamanhos Variáveis: Com o CIDR, os endereços IP estão contidos em prefixos de tamanhos variáveis. Isso permite uma divisão mais flexível do espaço de endereços e simplifica o roteamento em redes grandes.

Técnicas de Roteamento: Algoritmos complexos são usados para determinar a melhor rota para um pacote com base nos prefixos de roteamento disponíveis. Esses algoritmos são implementados em hardware especializado em roteadores comerciais.

Processo Automático de Agregação: A agregação de rotas é um processo automático que depende dos prefixos disponíveis na Internet. Ela reduz o número de entradas na tabela de roteamento, melhorando a eficiência do roteamento.

Maior Prefixo Combinado: Quando vários prefixos se sobrepõem, a regra do maior prefixo combinado é aplicada. Isso significa que a rota mais específica é usada para encaminhar o tráfego para o destino correto.

O CIDR é uma ferramenta fundamental para lidar com o crescimento contínuo da Internet e garantir o roteamento eficiente de pacotes em uma escala global.

O sistema de endereçamento em classes foi um dos primeiros métodos de alocação de endereços IP antes da introdução do CIDR. Aqui estão os principais pontos sobre o endereçamento em classes:

1. **Categorias de Classes**: Antes de 1993, os endereços IP eram divididos em cinco classes: A, B, C, D e E. Cada classe tinha um formato específico que determinava o número de redes e hosts que poderiam ser acomodados.
2. **Alocação Hierárquica**: O sistema de classes tinha uma estrutura hierárquica, mas os tamanhos dos blocos de endereço eram fixos. Isso resultou em um desperdício significativo de endereços, especialmente na classe B, que era muitas vezes maior do que o necessário para muitas organizações.
3. **Problema dos Três Ursos**: O problema dos três ursos refere-se à inadequação dos tamanhos de rede oferecidos pelas classes A, B e C para muitas organizações. Uma rede de classe B era muitas vezes muito grande, enquanto uma rede de classe C era muitas vezes muito pequena. Não havia um tamanho de rede "ideal" para muitas organizações.
4. **Sub-Redes e CIDR**: Para lidar com esses problemas, as sub-redes foram introduzidas para atribuir blocos de endereços com mais flexibilidade dentro de uma organização. Mais tarde, o CIDR foi introduzido para reduzir o tamanho da tabela de roteamento global e permitir uma alocação mais eficiente de endereços IP.
5. **Complexidade do Encaminhamento**: Com o CIDR, o encaminhamento tornou-se mais complexo, pois os prefixos de roteamento podiam ter tamanhos variáveis. Isso exigia algoritmos mais complexos para determinar a melhor rota para um pacote.
6. **Endereços Especiais**: Além das classes padrão, existem endereços IP especiais com significados específicos, como 0.0.0.0 (usado durante a inicialização), 255.255.255.255 (broadcast), e 127.x.x.x (loopback).

A transição do sistema de classes para o CIDR foi um marco importante na história da Internet, permitindo uma alocação mais eficiente de endereços IP e um roteamento mais flexível e escalável.

O NAT (Network Address Translation) é uma técnica usada para resolver o problema da escassez de endereços IP, permitindo que várias máquinas em uma rede privada compartilhem um único endereço IP público. Aqui está um resumo dos principais pontos sobre o NAT:

1. **Escassez de Endereços IP**: A crescente demanda por endereços IP levou à escassez deles. Com o NAT, um único endereço IP público pode ser compartilhado por várias máquinas em uma rede privada.
2. **Funcionamento do NAT**: Dentro da rede privada, cada computador tem um endereço IP exclusivo que é usado para roteamento interno. Quando um pacote sai da rede privada em direção à Internet, seu endereço IP interno é traduzido para o endereço IP público do NAT.
3. **Intervalos de Endereços Privados**: O NAT utiliza intervalos de endereços IP reservados para redes privadas, que não podem aparecer na Internet. Os intervalos comuns são 10.0.0.0/8, 172.16.0.0/12 e 192.168.0.0/16.
4. **Tradução de Portas**: Para mapear corretamente as conexões de saída, o NAT usa um conceito chamado tradução de portas. Cada conexão TCP ou UDP é associada a uma porta única na tabela de tradução do NAT.
5. **Problemas com o NAT**: Apesar de sua utilidade, o NAT enfrenta várias críticas. Ele viola o modelo arquitetônico do IP, dificulta a conectividade ponto a ponto, altera as características de rede e viola o princípio da independência entre camadas de protocolo.
6. **Limitações do NAT**: O NAT tem limitações em termos do número máximo de máquinas que podem ser mapeadas para um único endereço IP público. Isso ocorre devido ao espaço limitado de portas disponíveis.

Apesar das críticas e limitações, o NAT continua sendo amplamente utilizado em redes domésticas e de pequenas empresas como uma solução temporária para a escassez de endereços IP, especialmente enquanto a transição para o IPv6 está em andamento.

**5.6.3 IPV6**

1. **Contexto e Problemas do IPv4**: Inicia-se com a constatação de que o IPv4, embora tenha sido eficaz ao longo das décadas, enfrentava um problema crítico de esgotamento de endereços devido à sua popularidade. Mesmo com técnicas como CIDR e NAT, a escassez de endereços IPv4 era iminente.
2. **Solução Proposta: IPv6**: O texto destaca o IPv6 como a solução de longo prazo para o problema de esgotamento de endereços. Com endereços de 128 bits, o IPv6 oferece um número virtualmente ilimitado de endereços, resolvendo assim o problema de escassez.
3. **Dificuldades de Implementação**: Apesar de suas vantagens, o IPv6 enfrentou dificuldades significativas de implementação. O texto menciona que é um protocolo diferente da camada de rede e não se interliga facilmente com o IPv4. Além disso, muitas empresas e usuários não compreendem completamente as razões para adotá-lo.
4. **Características do IPv6**: O texto detalha as principais características do IPv6, incluindo:
   * Endereços de 128 bits
   * Simplificação do cabeçalho, reduzindo-o para sete campos em comparação com os 13 do IPv4
   * Melhor suporte para opções, tornando mais simples para os roteadores ignorarem opções não necessárias
   * Melhorias em segurança, incluindo autenticação e privacidade
   * Maior atenção à qualidade de serviço, especialmente com o crescimento da multimídia na Internet
5. **Comparação entre IPv4 e IPv6**: O texto compara os cabeçalhos do IPv4 e IPv6, destacando as diferenças significativas e as razões por trás delas, como a eliminação dos campos relacionados à fragmentação e o checksum no IPv6.
6. **Visão Futura**: O texto conclui com uma visão do potencial futuro do IPv6, destacando sua eficiência e flexibilidade como um protocolo rápido e capaz de oferecer um amplo espaço de endereços.

**Cabeçalhos de Extensão no IPv6:**

1. **Propósito dos Cabeçalhos de Extensão**: Os cabeçalhos de extensão no IPv6 foram introduzidos para fornecer informações adicionais de maneira eficiente, sendo opcionais e podendo ser criados conforme necessário.
2. **Tipos de Cabeçalhos de Extensão**: Existem seis tipos de cabeçalhos de extensão definidos no IPv6: hop-by-hop options, destination options, routing, fragmentation, authentication, e encrypted security payload.
3. **Formato dos Cabeçalhos de Extensão**: Alguns cabeçalhos têm formato fixo, enquanto outros possuem campos de comprimento variável codificados como tuplas (Tipo, Tamanho, Valor).

**Controvérsias e Desafios no Desenvolvimento do IPv6:**

1. **Tamanho do Endereço IPv6**: Houve debates sobre o tamanho ideal do endereço IPv6, resultando no uso de endereços de comprimento fixo de 16 bytes.
2. **Campo de Limite de Hops**: Discussões sobre limitar o número de hops a um máximo de 255 levaram à decisão de manter um campo de 8 bits para o limite de hops.
3. **Tamanho Máximo do Pacote**: Questões sobre o tamanho máximo do pacote foram resolvidas com a limitação de pacotes normais a 64 KB, enquanto o cabeçalho de extensão hop-by-hop permitiria jumbogramas maiores.
4. **Remoção do Checksum do IPv4**: Houve debates sobre a remoção do checksum do IPv4, com argumentos contra incluindo a redundância com checksums de camadas superiores e o desperdício de recursos.
5. **Compatibilidade com Hosts Móveis**: Não houve consenso sobre a inclusão de suporte explícito para hosts móveis no IPv6.
6. **Implementação de Mecanismos de Segurança**: Houve discussões sobre onde e como implementar mecanismos de segurança, considerando questões de confiança, controle e conformidade com regulamentações de exportação de criptografia.

5.6.7 ICMP - Internet Control Message Protocol:

O ICMP é fundamental para o funcionamento da Internet, pois fornece mensagens de controle e relatórios de erro. Ele opera encapsulado em pacotes IP e é usado para monitorar e diagnosticar problemas na rede. Algumas mensagens ICMP importantes incluem:

DESTINATION UNREACHABLE: Indica que o destino não pode ser alcançado, seja porque não existe ou porque o pacote não pode ser entregue.

TIME EXCEEDED: Informa que um pacote foi descartado devido ao esgotamento do contador de TTL, indicando possíveis problemas de roteamento.

PARAMETER PROBLEM: Indica que um valor inválido foi encontrado no cabeçalho do pacote, apontando possíveis bugs no software de rede.

SOURCE QUENCH: Antigamente usado para reduzir o tráfego excessivo, mas agora não é mais comumente utilizado devido à sua ineficácia no controle de congestionamento.

REDIRECT: Utilizado para instruir um host transmissor a atualizar sua rota para um destino específico.

ECHO e ECHO REPLY: Usados para testar a conectividade entre hosts, popularmente implementados no utilitário ping.

TIMESTAMP REQUEST e TIMESTAMP REPLY: Permitem medir o desempenho da rede, registrando o tempo de chegada e saída das mensagens.

ROUTER ADVERTISEMENT e ROUTER SOLICITATION: Facilitam a descoberta de roteadores vizinhos em uma rede local.

ARP - Address Resolution Protocol:

O ARP é utilizado para mapear endereços IP em endereços Ethernet em redes locais. Quando um host precisa enviar um pacote para outro na mesma rede, mas só conhece o endereço IP do destino, ele usa o ARP para descobrir o endereço Ethernet correspondente. Esse processo envolve a transmissão de uma solicitação ARP broadcast, para a qual o host de destino responde com seu endereço Ethernet. Essa resposta é armazenada em cache para futuras comunicações, otimizando o processo.

DHCP - Dynamic Host Configuration Protocol:

O DHCP é utilizado para configurar automaticamente os parâmetros de rede dos hosts, como endereço IP, máscara de sub-rede, gateway padrão, servidores DNS, entre outros. Ele funciona através de um servidor DHCP na rede, que atribui endereços IP a hosts que solicitam através de pacotes DHCP DISCOVER. Esses endereços são alocados temporariamente por um período de leasing, após o qual precisam ser renovados. O DHCP é amplamente utilizado em redes empresariais e domésticas para simplificar a configuração de dispositivos de rede.

5.6.8 O MPLS (MultiProtocol Label Switching) é uma tecnologia que adiciona rótulos aos pacotes para encaminhá-los de forma mais eficiente através de uma rede. Em vez de depender apenas dos endereços de destino IP, o MPLS utiliza rótulos para determinar a rota dos pacotes. Essa abordagem permite um encaminhamento mais rápido e flexível, adequado para a qualidade de serviço.

Os principais componentes do MPLS são:

1. **Rótulos**: São adicionados na frente de cada pacote e são usados como índices para tabelas de encaminhamento internas nos roteadores MPLS.
2. **Camada de rede e camada de enlace**: O MPLS fica entre o protocolo da camada de rede IP e o protocolo da camada de enlace (como o PPP). Ele não se encaixa perfeitamente em uma camada específica, às vezes sendo considerado um protocolo da "camada 2,5".
3. **Encaminhamento baseado em rótulos**: Os roteadores MPLS utilizam os rótulos para encaminhar os pacotes. Esses rótulos têm significado apenas localmente e são remapeados em cada hop para garantir a distinção dos pacotes.
4. **Classes de equivalência de encaminhamento (FEC)**: Os pacotes são agrupados em classes com base em seus rótulos e características, permitindo um encaminhamento eficiente e flexível.
5. **Pilhas de rótulos**: O MPLS pode operar em vários níveis, adicionando mais de um rótulo à frente de um pacote. Isso é chamado de pilha de rótulos e permite uma variedade de políticas de encaminhamento.
6. **Protocolos de controle**: São usados para estabelecer informações de encaminhamento de rótulos na rede. Esses protocolos são separados do encaminhamento de pacotes, permitindo uma variedade de abordagens para configurar as rotas.

Embora as ideias básicas do MPLS sejam simples, sua implementação envolve uma variedade de detalhes e casos de uso complexos. Essa tecnologia tem sido amplamente adotada por ISPs devido à sua capacidade de oferecer encaminhamento rápido e flexível, bem como suporte para qualidade de serviço. Para mais informações detalhadas, consulte as referências fornecidas.

O Protocolo de Roteamento de Estado de Enlace Aberto, conhecido como OSPF (Open Shortest Path First), é um protocolo de roteamento intra-domínio, também chamado de protocolo de gateway interior. Ele é amplamente utilizado na prática para roteamento dentro de sistemas autônomos (Autonomous Systems - AS) na Internet.

Antes do OSPF, os primeiros protocolos de roteamento intra-domínio utilizavam o algoritmo de vetor de distância, baseado no algoritmo de Bellman-Ford distribuído herdado da ARPANET. Um exemplo principal é o RIP (Routing Information Protocol), que ainda é utilizado atualmente. No entanto, o RIP possui algumas limitações, como o problema de contagem ao infinito e a convergência lenta, especialmente em redes maiores.

Em contraste, o OSPF foi padronizado em 1990 como um protocolo de estado de enlace para roteamento intra-domínio. Ele atendeu a uma série de requisitos e especificações:

1. **Publicação Aberta**: O algoritmo OSPF foi publicado na literatura aberta, o que significa que não era uma solução patenteada de uma empresa específica.
2. **Suporte a Diferentes Métricas de Distância**: O OSPF suporta uma variedade de métricas de distância, incluindo distância física, atraso, entre outras.
3. **Algoritmo Dinâmico**: É um algoritmo dinâmico que se adapta automaticamente às mudanças na topologia da rede.
4. **Roteamento Baseado no Tipo de Serviço**: Inicialmente, o OSPF foi projetado para suportar roteamento com base no tipo de serviço, mas esse recurso acabou sendo removido.
5. **Balanceamento de Carga**: O OSPF suporta balanceamento de carga, distribuindo o tráfego entre várias conexões, o que melhora o desempenho da rede.
6. **Suporte para Sistemas Hierárquicos**: O OSPF foi projetado para suportar sistemas hierárquicos, onde redes grandes podem ser divididas em áreas.
7. **Segurança**: O OSPF inclui mecanismos de segurança para evitar que informações de roteamento falsas sejam enviadas pelos roteadores.

O OSPF opera criando um grafo direcionado que representa a topologia da rede, onde cada nó representa um roteador e cada arco representa uma conexão entre roteadores ou redes. O algoritmo de estado de enlace é usado para calcular o caminho mais curto de um roteador para todos os outros nós na rede.

O OSPF suporta a divisão de um AS em áreas numeradas, onde cada área é uma rede ou um conjunto de redes contíguas. Isso ajuda na escalabilidade do roteamento e na redução do tráfego de informações de roteamento. Cada AS também tem uma área de backbone (área 0), que é responsável por conectar todas as outras áreas.

Existem diferentes tipos de roteadores no OSPF, como roteadores internos, roteadores de borda de área, roteadores de backbone e roteadores de limite de AS, cada um desempenhando um papel específico no roteamento e na troca de informações de roteamento.

Em resumo, o OSPF é um protocolo de roteamento intra-domínio robusto e altamente escalável, projetado para atender às necessidades das redes modernas, oferecendo eficiência, segurança e flexibilidade.

**BGP**

1. **Diferenças entre protocolos de roteamento intra e interdomínio:** Enquanto OSPF e IS-IS são usados dentro de um único sistema autônomo (AS), o BGP é usado entre ASs. Isso ocorre porque os objetivos de roteamento dentro de um domínio são diferentes dos objetivos entre domínios.
2. **Políticas de roteamento interdomínio:** O BGP permite a implementação de políticas de roteamento que refletem considerações políticas, de segurança e econômicas. Essas políticas incluem restrições como não transportar tráfego comercial em redes educacionais ou evitar rotas que passem por determinadas regiões por motivos políticos.
3. **Serviços de trânsito e peering:** ASs podem comprar serviços de trânsito de outros ASs para alcançar destinos na Internet ou estabelecer acordos de peering para trocar tráfego diretamente, sem custos monetários. Esses acordos são importantes para otimizar o tráfego e reduzir custos.
4. **Multihoming:** Algumas redes de empresas se conectam a vários ISPs para melhorar a confiabilidade. Nesses casos, o BGP é usado para informar a outros ASs quais endereços devem ser alcançados por meio de quais enlaces de ISP.
5. **Funcionamento do BGP:** O BGP é um protocolo de vetor de caminho que mantém informações sobre o caminho utilizado para cada rota. Os roteadores BGP se comunicam usando conexões TCP, e as rotas são propagadas entre ASs. Dentro de um AS, o BGP interno (iBGP) é usado para propagar rotas.
6. **Seleção de rotas no BGP:** Cada roteador BGP escolhe a melhor rota com base em políticas específicas configuradas pelo ISP. Fatores como custo, tipo de relação (cliente, peering, trânsito) e caminho do AS são considerados na escolha da rota.
7. **Complexidade do BGP:** O protocolo BGP é complexo devido à variedade de políticas de roteamento e à necessidade de coordenar escolhas de rota entre os roteadores dentro de um AS. A RFC 4271 e outras RFCs relacionadas fornecem detalhes sobre o funcionamento do BGP.

Esses são os pontos essenciais abordados no texto sobre o BGP e seu papel no roteamento interdomínio na Internet.

multicast na Internet, destacando os seguintes pontos principais:

1. **Suporte a Multicast pelo IP:** O multicast permite a comunicação de um para muitos, onde um processo pode enviar dados para um grupo de receptores simultaneamente. Isso é útil para aplicações como streaming ao vivo, atualizações de programa e conferências digitais.
2. **Endereços IP de Classe D:** O IP suporta multicast usando endereços de classe D. Cada endereço de classe D identifica um grupo de hosts, e mais de 250 milhões de grupos podem existir ao mesmo tempo.
3. **Multicast na Rede Local:** A faixa de endereços IP 224.0.0.0/24 é reservada para multicast na rede local. Nesse caso, os pacotes são transmitidos simplesmente na LAN usando endereços de multicast, e os roteadores não encaminham os pacotes. Exemplos de endereços de multicast local incluem 224.0.0.1 para todos os sistemas em uma LAN e 224.0.0.2 para todos os roteadores em uma LAN.
4. **Protocolo de Gerenciamento de Grupo da Internet (IGMP):** Para redes com membros de grupo em redes diferentes, os roteadores de multicast precisam saber quais hosts são membros de um grupo. O IGMP é usado para os hosts se juntarem a grupos específicos e para os roteadores multicast descobrirem quais hosts são membros de quais grupos.
5. **Protocolos de Roteamento Multicast:** Diversos protocolos de roteamento multicast podem ser usados para criar árvores de multicast que oferecem caminhos dos transmissores para todos os membros do grupo. O Protocol Independent Multicast (PIM) é o protocolo principal usado dentro de um AS, com variantes como Dense Mode PIM e Sparse Mode PIM, dependendo da distribuição dos membros do grupo na rede.
6. **Extensões de Multicast ao BGP:** Quando os membros do grupo estão em mais de um AS, são necessárias extensões de multicast ao BGP ou túneis para criar rotas multicast.

Esses pontos abordados no texto fornecem uma compreensão básica do multicast na Internet, desde o suporte básico no IP até os protocolos e técnicas usados para implementá-lo em diferentes contextos de rede.

**IP**

1. **Desafios do Endereçamento IP Tradicional:** O texto começa destacando como o sistema de endereçamento IP tradicional torna difícil para os usuários permanecerem conectados quando estão longe de casa, pois os endereços IP são associados a uma rede específica.
2. **Objetivos da Solução de IP Móvel:** Um grupo de trabalho foi estabelecido para encontrar uma solução para permitir que hosts móveis usem seu endereço IP doméstico de qualquer lugar, sem exigir mudanças nos hosts fixos ou roteadores.
3. **Solução Escolhida:** A solução escolhida envolve a criação de um "agente de origem" em cada local que deseja permitir que seus usuários viajem. Quando um host móvel aparece em um local externo, ele obtém um novo endereço IP nesse local e informa ao agente de origem onde ele está. O agente de origem então encaminha os pacotes para o host móvel através de um túnel.
4. **Protocolos e Mecanismos:** O texto discute o uso de mensagens ICMP, DHCP e ARP para permitir que os dispositivos móveis descubram sua localização atual e se comuniquem com seus agentes de origem. O tunelamento é usado para encaminhar pacotes entre o agente de origem e o host móvel.
5. **Desafios e Soluções Adicionais:** São abordados desafios como o roteamento triangular e a filtragem de acesso por ISPs, além de soluções como a otimização de rota e o uso de protocolos de autenticação criptográfica para garantir a segurança.
6. **IPv6 e Mobilidade:** O texto menciona que os protocolos de mobilidade para o IPv6 baseiam-se nos fundamentos do IPv4 e discute como o IPv6 móvel é definido na RFC 3775. Também menciona a mobilidade da rede definida para o IPv6 na RFC 3963.

Essencialmente, o IP móvel permite que dispositivos permaneçam conectados à Internet enquanto se movem entre diferentes locais, garantindo que eles possam usar seu endereço IP doméstico e manter a conectividade sem interrupções.