**Capítulo 3- A camada de Enlace de dados**

**3.1 Questões de Projeto da camada de enlace de dados**

* A camada de enlace de dados tem diversas funções, incluindo fornecer uma interface de serviço à camada de rede, lidar com erros de transmissão e regular o fluxo de dados.
* Para alcançar esses objetivos, a camada de enlace de dados encapsula pacotes da camada de rede em quadros para transmissão, cada um contendo um cabeçalho, um campo de carga útil e um final de quadro.
* O gerenciamento de quadros é central nas atividades da camada de enlace de dados.
* Embora este capítulo trate especificamente da camada de enlace de dados, os princípios como controle de erros e controle de fluxo são encontrados em protocolos de transporte e outros protocolos, pois a confiabilidade é um objetivo geral.
* Em algumas redes, essas funções estão nas camadas superiores, com a camada de enlace de dados realizando uma tarefa mínima. No entanto, os princípios são quase idênticos independentemente do local onde são encontrados. A camada de enlace de dados fornece uma oportunidade para examiná-los em detalhes.

**3.1.1 serviços oferecidos à camada de rede**

A função da camada de enlace de dados é transferir dados da camada de rede da máquina de origem para a máquina de destino, utilizando serviços oferecidos à camada de rede.

* Três possíveis serviços oferecidos pela camada de enlace de dados são:
  1. Serviço não orientado a conexões sem confirmação: A máquina de origem envia quadros independentes à máquina de destino sem receber confirmação de recebimento. Exemplo: Ethernet.
  2. Serviço não orientado a conexões com confirmação: Cada quadro enviado é confirmado individualmente, permitindo ao transmissor saber se o quadro chegou corretamente. Exemplo: Padrão 802.11 (WiFi).
  3. Serviço orientado a conexões com confirmação: Estabelece conexões lógicas antes da transmissão de dados e confirma cada quadro enviado, garantindo sua recepção e ordem correta. Exemplo: Protocolos de comunicação serial.
* Oferecer confirmação na camada de enlace de dados é uma otimização, não uma exigência, mas é útil em canais não confiáveis, como sistemas sem fio, para corrigir erros mais rapidamente.
* O serviço mais sofisticado é o orientado a conexões, onde uma conexão é estabelecida antes da transferência de dados, garantindo confiabilidade e ordem na recepção dos quadros.
* A transferência de dados em um serviço orientado a conexões passa por três fases distintas: estabelecimento da conexão, transmissão dos quadros e encerramento da conexão.

**3.1.2 EnquadramentoParte superior do formulário**

* A camada de enlace de dados utiliza o serviço fornecido pela camada física para transferir dados da máquina de origem para a máquina de destino.
* A camada física aceita um fluxo de bits brutos e tenta entregá-lo ao destino, adicionando redundância aos sinais para reduzir a taxa de erros, especialmente em enlaces com ruído.
* No entanto, o fluxo de bits recebido pela camada de enlace de dados pode conter erros, como bits com valores diferentes ou número diferente de bits recebidos em relação aos transmitidos, exigindo detecção e, se necessário, correção de erros.
* Uma estratégia comum adotada pela camada de enlace de dados é dividir o fluxo de bits em quadros distintos e calcular um checksum para cada quadro, incluindo-o no quadro transmitido para detecção de erros.
* Existem diferentes métodos de enquadramento para dividir o fluxo de bits em quadros, como contagem de caracteres, bytes de flag com inserção de bytes, flags iniciais e finais com inserção de bits e violações de codificação da camada física.
* Muitos protocolos de enlace de dados utilizam uma combinação desses métodos para delimitar os quadros de forma eficaz, garantindo uma transmissão confiável e eficiente.

3.1.3 Controle de erros

Para garantir uma entrega confiável dos quadros na camada de rede de destino e na ordem apropriada, alguns procedimentos são adotados:

1. **Feedback do receptor**: Normalmente, o protocolo solicita que o receptor retorne quadros de controle especiais com confirmações positivas ou negativas sobre os quadros recebidos. Isso permite que o transmissor saiba se os quadros foram entregues com sucesso ou se precisam ser retransmitidos.
2. **Introdução de timers**: O transmissor inicia um timer quando envia um quadro. Se o tempo definido pelo timer expirar antes de receber uma confirmação do receptor, o transmissor assume que houve um problema e retransmite o quadro.
3. **Atribuição de números de sequência**: Para evitar que o receptor aceite o mesmo quadro múltiplas vezes em caso de retransmissão, são atribuídos números de sequência aos quadros transmitidos. Isso permite que o receptor distinga entre retransmissões e quadros originais.

Essas estratégias são essenciais para garantir a entrega confiável dos dados na camada de rede de destino, minimizando o risco de perdas e garantindo a ordem correta dos quadros. O gerenciamento dos timers e dos números de sequência é uma parte fundamental das responsabilidades da camada de enlace de dados e de camadas superiores do modelo OSI.

**3.1.4 Controle de fluxo**

O controle de fluxo na camada de enlace de dados é crucial para garantir que o transmissor não sobrecarregue o receptor com dados mais rapidamente do que ele pode processar. Existem duas abordagens comuns para o controle de fluxo:

1. **Controle de fluxo baseado em feedback**: Nessa abordagem, o receptor envia feedback ao transmissor para informar sua capacidade de receber dados. O transmissor então ajusta sua taxa de transmissão com base nessas informações. Isso pode ser feito de várias maneiras, como confirmando explicitamente a recepção de quadros ou indicando a quantidade de espaço de buffer disponível no receptor.
2. **Controle de fluxo baseado na velocidade**: Nessa abordagem, o protocolo possui mecanismos internos para limitar a velocidade com que os dados são enviados, independentemente do feedback do receptor. Essa abordagem é menos comum na camada de enlace de dados e é mais frequentemente implementada na camada de transporte.

Geralmente, os esquemas de controle de fluxo baseados em feedback são preferidos, pois permitem uma adaptação dinâmica à capacidade do receptor. No entanto, as implementações modernas de hardware de camada de enlace, como placas de interface de rede (NICs), muitas vezes são projetadas para lidar com dados rapidamente o suficiente para evitar problemas de sobrecarga no receptor.

Os detalhes específicos dos esquemas de controle de fluxo podem variar, mas geralmente seguem o princípio básico de permitir que o transmissor envie dados somente quando o receptor estiver pronto para recebê-los. Isso pode envolver regras explícitas sobre quando o transmissor pode enviar quadros ou informações sobre a quantidade de dados que o receptor pode receber.

**3.2 Detecção e correção de erros**

Na transmissão de dados, erros podem ocorrer devido a várias razões, como ruído no canal de comunicação. Para lidar com esses erros, os projetistas de redes desenvolveram duas estratégias básicas: detecção de erros e correção de erros.

1. **Detecção de erros**: Nessa estratégia, são adicionadas informações redundantes aos dados transmitidos de modo que o receptor possa detectar a ocorrência de erros. No entanto, o receptor não consegue corrigir os erros por si só. Em vez disso, ele reconhece que ocorreu um erro e solicita uma retransmissão dos dados. Isso é útil em canais com taxas de erro baixas ou moderadas, onde a retransmissão ocasional de dados com erros não compromete significativamente o desempenho.
2. **Correção de erros**: Nessa estratégia, informações redundantes são adicionadas aos dados de modo que o receptor possa não apenas detectar a ocorrência de erros, mas também corrigi-los sem a necessidade de uma retransmissão. Isso é especialmente útil em canais com taxas de erro mais altas, como em enlaces sem fio, onde retransmissões frequentes podem degradar significativamente o desempenho da rede.

Existem diferentes tipos de erros que podem ocorrer, como erros isolados de bits, erros em rajadas, ou até mesmo erros em que o local do erro é conhecido (canal de apagamento). Cada tipo de erro apresenta seus próprios desafios em termos de detecção e correção.

Os códigos de detecção e correção de erros são fundamentais para garantir a confiabilidade da transmissão de dados em redes de computadores. Embora esses códigos possam ser bastante complexos e baseados em princípios matemáticos, muitos protocolos de rede utilizam códigos padronizados para garantir a integridade dos dados transmitidos.

**3.2.1 códigos de correção de erros**

Os códigos de correção de erros desempenham um papel crucial na garantia da integridade das informações transmitidas em sistemas de comunicação. Vamos revisar brevemente os quatro tipos mencionados:

1. **Códigos de Hamming**: Esses códigos adicionam bits de verificação aos dados transmitidos para permitir a detecção e correção de erros. Eles são lineares e sistemáticos, o que significa que os bits de verificação são calculados de forma linear em relação aos bits de dados. A distância de Hamming é fundamental para a detecção e correção de erros nesses códigos.
2. **Códigos de Convolução Binários**: Esses códigos não são de bloco, mas sim baseados em uma sequência de entrada e saída. Eles são amplamente utilizados em sistemas de comunicação, como GSM, comunicações por satélite e redes sem fio. O algoritmo de Viterbi é frequentemente utilizado para decodificação, especialmente para códigos de convolução com comprimentos de restrição curtos.
3. **Códigos de Reed-Solomon**: Esses códigos são eficazes na correção de erros, especialmente erros em rajada, e são amplamente utilizados em CDs, DVDs, comunicações via satélite, entre outros. Eles operam sobre símbolos de bits e são baseados em polinômios sobre campos finitos.
4. **Códigos de Verificação de Paridade de Baixa Densidade (LDPC)**: Inventados por Robert Gallagher, esses códigos são conhecidos por sua eficácia na correção de erros e são utilizados em uma variedade de aplicações, incluindo redes de linha de energia, Ethernet de 10 Gbps e difusão de vídeo digital. Eles são representados por matrizes com uma baixa densidade de uns, o que contribui para suas excelentes capacidades de correção de erro.

Cada tipo de código tem suas próprias características e aplicações específicas, e a escolha do código depende das necessidades do sistema de comunicação em questão.

**3.2.2 Correção de erros**

Os códigos de detecção de erros são essenciais em comunicações, especialmente em ambientes propensos a erros, como enlaces sem fio. Eles ajudam a garantir a integridade dos dados transmitidos, permitindo a detecção e, em alguns casos, a correção de erros.

Vamos analisar os três principais códigos de detecção de erros mencionados no texto:

1. **Paridade**: Um bit de paridade é adicionado aos dados, de modo que o número total de bits 1 na palavra de código seja par (ou ímpar). Isso pode ser feito calculando o bit de paridade como a soma de módulo 2 ou a operação XOR dos bits de dados. A paridade pode detectar erros de único bit, mas não é eficaz para erros em rajada.
2. **Checksums**: São somas acumuladas dos bits de dados da mensagem. Um exemplo é o checksum de 16 bits da Internet. Os checksums são colocados no final da mensagem e podem detectar vários tipos de erros, mas podem não ser robustos o suficiente em algumas situações.
3. **Verificações de Redundância Cíclica (CRCs)**: Baseiam-se na representação dos dados como polinômios binários. Um polinômio gerador é acordado entre o transmissor e o receptor. O transmissor calcula um CRC e o anexa aos dados. O receptor verifica se o CRC é divisível pelo polinômio gerador. Os CRCs são amplamente utilizados na camada de enlace de dados e são altamente eficazes na detecção de erros em rajada.

Os CRCs têm se destacado como a escolha predominante para detecção de erros em comunicações devido à sua eficácia e eficiência. Apesar de exigirem cálculos mais complexos, são implementados em hardware de forma eficiente e são capazes de detectar uma ampla gama de erros, incluindo erros em rajada.

**3.3 O protocolos básicos de enlace de dados**

Os protocolos básicos de enlace de dados descritos nesta seção introduzem conceitos fundamentais para a comunicação de dados em redes de computadores.

**Implementação das Camadas**: Na implementação, são consideradas três camadas independentes: física, de enlace de dados e de rede. O hardware dedicado, como a placa de interface de rede (NIC), lida com a camada física e parte da camada de enlace de dados, enquanto o restante da camada de enlace de dados e a camada de rede operam na CPU principal como parte do sistema operacional.

1. **Suposições Fundamentais**:
   * Uma máquina deseja enviar um fluxo de dados longo a outra máquina utilizando um serviço confiável e orientado a conexões.
   * Supõe-se que não ocorram panes nos computadores.
   * O pacote recebido pela camada de enlace de dados consiste em dados puros, e o cabeçalho do pacote não é relevante para essa camada.
2. **Encapsulamento em Quadros**: Quando um pacote é aceito pela camada de enlace de dados, ele é encapsulado em um quadro, ao qual é adicionado um cabeçalho e um final de enlace de dados. O quadro é então transmitido à camada de enlace de dados da máquina de destino.
3. **Tratamento de Erros**: Quando um quadro chega ao receptor, o checksum é recalculado para verificar se houve erro de transmissão. Se o checksum estiver incorreto, a camada de enlace de dados é informada; caso contrário, o quadro é entregue à camada de rede.
4. **Separação de Protocolos**: É mantida uma separação estrita entre os protocolos de rede e de enlace de dados para permitir que evoluam independentemente. A camada de rede não precisa conhecer os detalhes do protocolo de enlace de dados ou o formato do quadro.
5. **Definições e Funções Comuns**: São apresentadas diversas definições e funções comuns aos protocolos discutidos, incluindo estruturas de dados para representar pacotes e quadros, além de funções para lidar com a comunicação entre as camadas.

Esses conceitos básicos são fundamentais para compreender o funcionamento dos protocolos de enlace de dados e formam a base para protocolos mais complexos e sofisticados.

**3.3.2 O protocolo simplex**

O protocolo simplex sem restrições, chamado de "utopia", é um exemplo extremamente simplificado que não lida com problemas típicos encontrados em comunicações reais, como controle de fluxo e correção de erros.

**Características do Protocolo**:

* + Transmissão unidirecional: Os dados são transmitidos em apenas um sentido, sem considerar a comunicação bidirecional.
  + Ausência de restrições: Não há preocupação com possíveis problemas, como erros de transmissão, perda de quadros, capacidade do buffer, tempo de processamento, entre outros.

1. **Funcionamento**:
   * Transmissor: O transmissor opera em um loop infinito, enviando dados o mais rápido possível. Ele busca um pacote da camada de rede, cria um quadro e o transmite ao destino. Como não há controle de fluxo nem detecção/correção de erros, apenas o campo de informações do quadro é usado.
   * Receptor: O receptor espera indefinidamente por um quadro não danificado. Quando um quadro chega, a função de espera por eventos retorna, indicando a chegada do quadro. O quadro é então removido do buffer do hardware e os dados são repassados à camada de rede.
2. **Limitações**:
   * Ausência de controle de fluxo: Não há mecanismos para regular o fluxo de dados entre o transmissor e o receptor, o que pode levar a problemas de congestionamento.
   * Ausência de detecção/correção de erros: O protocolo não possui mecanismos para detectar ou corrigir erros de transmissão, o que pode resultar na entrega de dados incorretos ao destino.
   * Sem confirmações: Não há confirmações de recebimento dos quadros, o que significa que o transmissor não sabe se os dados foram recebidos corretamente pelo receptor.

Em resumo, o protocolo "utopia" é uma simplificação extrema que não representa as complexidades e desafios reais enfrentados em redes de computadores. Ele serve como um exemplo didático para ilustrar os conceitos básicos de transmissão de dados, mas não é adequado para aplicações práticas devido à sua falta de recursos para lidar com problemas comuns de comunicação.

**3.3.3 protocolo apresentado é conhecido como "stop-and-wait**

O protocolo apresentado é conhecido como "stop-and-wait" e visa evitar que o transmissor sobrecarregue o receptor enviando dados mais rapidamente do que ele é capaz de processar. Aqui estão os principais pontos desse protocolo:

1. **Características do Protocolo**:
   * Transmissão unidirecional: Assim como no protocolo "utopia", os dados são transmitidos em apenas um sentido, do transmissor para o receptor.
   * Canal livre de erros: O protocolo presume que o canal de comunicação não apresenta erros, portanto não há necessidade de mecanismos de detecção ou correção de erros.
   * Controle de fluxo simples: O receptor envia um quadro de confirmação de volta ao transmissor após receber e processar com sucesso um quadro de dados.
2. **Funcionamento**:
   * Transmissor (**sender2**): O transmissor extrai um pacote da camada de rede, cria um quadro com esses dados e o transmite ao destino. Em seguida, espera pela chegada de um quadro de confirmação antes de prosseguir para o próximo pacote.
   * Receptor (**receiver2**): O receptor espera pela chegada de um quadro de dados. Quando ele chega, o receptor o processa, passa os dados à camada de rede e envia um quadro de confirmação de volta ao transmissor.
3. **Alternância de fluxo**: O protocolo stop-and-wait impõe uma rígida alternância de fluxo, onde o transmissor envia um quadro de cada vez e aguarda a confirmação do receptor antes de enviar o próximo. Isso garante que o receptor não seja sobrecarregado com dados que não consegue processar.
4. **Meio físico half-duplex**: Como há uma alternância clara entre o envio de dados e a confirmação, um canal físico half-duplex (que permite a comunicação em apenas uma direção por vez) é suficiente para esse protocolo.

No geral, o stop-and-wait é um protocolo simples e eficaz para controle de fluxo em canais livres de erros, garantindo que o transmissor não sobrecarregue o receptor com dados não processados. No entanto, ele pode ser lento em comparação com outros protocolos mais avançados, pois exige uma espera pelo retorno da confirmação antes de enviar o próximo pacote.

**3.4 Protocolo da Janela Deslizante**

Os protocolos de janela deslizante são utilizados quando há a necessidade de transmitir dados em ambos os sentidos de forma eficiente e confiável. Vamos analisar os conceitos e características desses protocolos:

1. **Canal de comunicação**: Em situações práticas, é comum utilizar o mesmo circuito para transmitir dados em ambos os sentidos. Isso significa que os quadros de dados enviados de A para B são misturados com os quadros de confirmação enviados de B para A.
2. **Piggybacking**: Uma técnica utilizada nos protocolos de janela deslizante é o piggybacking, onde o receptor espera até que a camada de rede envie o próximo pacote antes de enviar uma confirmação. A confirmação é então incluída no próximo quadro de dados que está sendo enviado, aproveitando melhor a largura de banda disponível.
3. **Complexidade do Piggybacking**: Apesar dos benefícios do piggybacking, introduz uma complicação relacionada ao tempo de espera para incluir a confirmação no próximo quadro de dados. Se o receptor esperar muito tempo, pode levar à retransmissão do quadro, invalidando o processo de confirmação.
4. **Protocolos de Janela Deslizante**: Existem diferentes protocolos de janela deslizante, onde cada quadro enviado contém um número de sequência. Os protocolos mantêm conjuntos de números de sequência correspondentes aos quadros que podem ser enviados ou aceitos.
5. **Janela de Transmissão e Recebimento**: Tanto o transmissor quanto o receptor mantêm uma janela de transmissão e recebimento, respectivamente. Essas janelas representam o conjunto de quadros que podem ser enviados ou aceitos em um determinado momento.
6. **Armazenamento de Quadros Não Confirmados**: O transmissor deve manter em memória todos os quadros não confirmados, o que pode exigir buffers significativos. Se a janela atingir seu tamanho máximo, a camada de enlace de dados do transmissor será desativada até que outro buffer esteja disponível.
7. **Descarte de Quadros Fora da Janela**: Qualquer quadro recebido fora da janela de recebimento do receptor será descartado. Somente os quadros dentro da janela serão processados e passados para a camada de rede.

Os protocolos de janela deslizante são eficientes e garantem a entrega confiável de dados em ambas as direções, utilizando estratégias como piggybacking para otimizar o uso da largura de banda disponível. No entanto, eles também introduzem complexidade adicional devido ao gerenciamento das janelas de transmissão e recebimento, bem como ao armazenamento de quadros não confirmados.

**3.4.1 O protocolo de janela deslizante de um bit**

O protocolo de janela deslizante de um bit, representado pelo Quadro 3.5, é um exemplo simples que utiliza a técnica stop-and-wait, onde o transmissor envia um quadro e aguarda sua confirmação antes de enviar o próximo. Vamos analisar os principais aspectos desse protocolo:

1. **Variáveis de Controle**: O protocolo define as variáveis **next\_frame\_to\_send** e **frame\_expected**, que indicam respectivamente o próximo quadro a ser enviado pelo transmissor e o próximo quadro esperado pelo receptor. Ambas variam entre 0 e 1.
2. **Envio e Recebimento de Quadros**: O transmissor envia um quadro contendo o pacote atual da camada de rede, aguarda sua confirmação e reenvia em caso de timeout. O receptor recebe os quadros, verifica se estão corretos e passa os dados para a camada de rede. Além disso, o receptor envia confirmações para os quadros recebidos.
3. **Tratamento de Situações Patológicas**: O protocolo é capaz de lidar com situações em que ocorrem timeouts prematuros ou envio de pacotes duplicados. Por exemplo, se o transmissor enviar várias cópias do mesmo quadro devido a timeouts prematuros, o receptor rejeitará todas, exceto a primeira.
4. **Sincronização**: O protocolo pode entrar em uma situação anormal se os dois lados começarem a comunicação simultaneamente. Nesse caso, os primeiros quadros podem se cruzar, levando a cópias desnecessárias de quadros. Isso pode ocorrer mesmo quando um lado começa antes do outro e pode resultar em desperdício de largura de banda.

Em resumo, o protocolo de janela deslizante de um bit é uma forma simples e eficiente de garantir a transmissão confiável de dados em uma rede. No entanto, ele pode enfrentar desafios em situações de sincronização ou em caso de timeouts prematuros, que podem levar ao envio desnecessário de quadros.

**3.4.2 o protocolo de controle de erro conhecido como go-back-n.**

texto descreve o protocolo de controle de erro conhecido como **go-back-n**. Esse protocolo é uma extensão do protocolo stop-and-wait, onde o transmissor pode enviar vários quadros antes de receber confirmações para cada um deles.

A principal ideia do go-back-n é permitir que o transmissor envie uma janela de quadros antes de ser bloqueado para aguardar confirmações. Isso aumenta a eficiência do uso da largura de banda em comparação com o stop-and-wait.

No entanto, quando ocorre um erro em um dos quadros dentro da janela de transmissão, todos os quadros subsequentes a ele são descartados e retransmitidos, daí o termo "go back". Isso ocorre porque o receptor descarta os quadros danificados e não envia confirmações para eles, o que faz com que o transmissor reenvie todos os quadros a partir do último confirmado.

O texto também discute a questão do timeout e da janela de transmissão, explicando como calcular o tamanho ideal da janela para otimizar o uso da largura de banda. Além disso, aborda estratégias para lidar com erros, como retransmissão seletiva e confirmações cumulativas.

No código fornecido, é apresentada uma implementação do protocolo go-back-n em C, incluindo funções para enviar dados, lidar com a chegada de quadros, tratamento de erros e gerenciamento de timeouts. O código usa uma abordagem de lista encadeada para gerenciar os timers dos quadros pendentes.

**3.4.3 um protocolo que utiliza retransmissão seletiva**

1. **Retransmissão Seletiva:** Em contraste com o protocolo go-back-n, que retransmite todos os quadros a partir do primeiro quadro não confirmado, o protocolo descrito permite a retransmissão seletiva de quadros individuais.
2. **Janela Deslizante:** Tanto o transmissor quanto o receptor mantêm uma janela de números de sequência aceitáveis. Isso permite que o transmissor envie vários quadros antes de receber confirmações, melhorando a eficiência da comunicação.
3. **Buffer e Bit de Chegada:** O receptor mantém um buffer para cada número de sequência dentro de sua janela aceitável. Cada buffer tem um bit associado para indicar se o buffer está cheio ou vazio.
4. **Tratamento de Quadros Recebidos:** Quando um quadro chega, o receptor verifica se ele se encaixa na janela aceitável. Se sim, o quadro é aceito e armazenado no buffer correspondente. Os quadros são repassados para a camada de rede apenas quando todos os quadros de números de sequência inferiores já foram entregues.
5. **Retransmissão e Controle de Janela:** O transmissor retransmite apenas quadros individuais que tenham expirado seu timer. A janela deslizante é usada para controlar quais quadros podem ser enviados.
6. **Evitando Conflitos e Problemas de Timeout:** O protocolo usa técnicas para evitar conflitos, garantindo que o receptor possa distinguir entre retransmissões e novos quadros. Também aborda o problema de determinar qual quadro provocou um timeout quando vários quadros estão pendentes.
7. **Estratégia de Timeout e NAKs:** O protocolo utiliza timeouts e NAKs (Negative Acknowledgments) para lidar com erros e garantir a entrega correta dos quadros. Um timer auxiliar é usado para garantir uma taxa mínima de confirmações, e NAKs são enviadas para solicitar retransmissões quando necessário.
8. **Ajuste de Parâmetros e Eficiência:** O protocolo adapta seus parâmetros, como o tamanho da janela e os timeouts, com base nas características da rede para melhorar a eficiência de comunicação.

Em resumo, o protocolo de retransmissão seletiva descrito oferece uma abordagem mais eficiente e robusta para lidar com erros em comunicações de rede, permitindo a recuperação seletiva de quadros perdidos ou danificados.

**3.5 exemplos de protocolos de enlace de dados**

Protocolos de enlace de dados são essenciais para garantir a comunicação confiável entre dispositivos em uma rede. Aqui estão dois exemplos de protocolos de enlace de dados utilizados em diferentes situações:

1. **PPP (Point-to-Point Protocol):**
   * O PPP é um protocolo padrão utilizado para estabelecer uma conexão ponto a ponto entre dois dispositivos, como um computador e um roteador, através de uma linha de comunicação serial.
   * É amplamente utilizado em diversas tecnologias de comunicação, incluindo linhas discadas, linhas alugadas, modems a cabo e fibra óptica SONET.
   * O PPP é definido na RFC 1661 e mais detalhadamente na RFC 1662, entre outras RFCs.
   * Ele oferece recursos como autenticação, detecção e correção de erros, compressão de dados e suporte a protocolos de rede como TCP/IP.
   * Em enlaces SONET e ADSL, o PPP é aplicado de maneiras diferentes, mas sua flexibilidade e capacidade de adaptação o tornam uma escolha comum para uma variedade de cenários de rede.
2. **Protocolos Específicos de ADSL e SONET:**
   * Em enlaces ADSL (Asymmetric Digital Subscriber Line), que conectam usuários finais à Internet através da rede telefônica, e em enlaces SONET (Synchronous Optical Networking), amplamente utilizados em redes de longa distância, podem ser aplicados protocolos específicos para otimizar a comunicação nessas tecnologias.
   * Embora o PPP seja comumente usado nessas conexões, outros protocolos podem ser empregados para fins específicos, como otimização de largura de banda, gerenciamento de tráfego e suporte a funcionalidades avançadas dessas tecnologias.
   * Em geral, os protocolos específicos de ADSL e SONET são projetados para oferecer desempenho máximo e confiabilidade em ambientes de rede específicos.

**3.5.1 pacotes sobre sonet**

O transporte de pacotes sobre SONET envolve o uso do protocolo PPP (Point-to-Point Protocol) para fornecer o mecanismo de enquadramento necessário para distinguir os pacotes do fluxo contínuo de bits transmitidos. descrição do processo:

1. **Enquadramento com PPP:**
   * O PPP atua como um mecanismo de enquadramento para pacotes transportados sobre enlaces SONET. Ele delineia de forma não ambígua o início e o fim de cada quadro e também lida com a detecção de erros.
   * Além disso, o PPP oferece um protocolo de controle de enlace, chamado LCP (Link Control Protocol), usado para ativar, testar, negociar opções e desativar linhas de forma controlada quando não são mais necessárias.
   * O PPP também oferece um método para negociar opções de camada de rede de forma independente do protocolo da camada utilizada, através de protocolos de controle de rede (NCP - Network Control Protocol) específicos para cada camada de rede aceita.
2. **Formato do Quadro PPP:**
   * O formato do quadro PPP é semelhante ao formato do quadro HDLC (High-level Data Link Control), com algumas diferenças significativas.
   * Todos os quadros PPP começam e terminam com o byte de flag padrão do HDLC (0x7E), facilitando a detecção do início e do fim dos quadros.
   * O campo Endereço é sempre definido como 11111111, indicando que todos os dispositivos devem aceitar o quadro.
   * O campo Controle indica o tipo de quadro e, na configuração padrão, é definido como 00000011 para indicar um quadro não numerado.
   * O campo Protocolo informa o tipo de pacote contido na carga útil, como pacotes IP, e permite a negociação de um tamanho de campo reduzido através do LCP.
   * O campo Carga útil contém os dados do pacote, com um comprimento variável e um checksum no final para detecção de erros.
3. **Configuração do Enlace PPP sobre SONET:**
   * Antes de transportar pacotes PPP sobre enlaces SONET, o enlace PPP precisa ser estabelecido e configurado.
   * O processo envolve a negociação de opções através do LCP, autenticação mútua das partes, se desejado, e a configuração da camada de rede através de pacotes NCP específicos para cada protocolo de rede admitido.
   * Quando o enlace alcança o estado OPEN, os pacotes IP podem ser transportados em quadros PPP pela linha SONET.

Esse processo garante que os pacotes possam ser transmitidos de forma confiável e eficiente sobre enlaces SONET, aproveitando as capacidades do protocolo PPP para fornecer enquadramento, detecção de erros, autenticação e configuração da camada de rede.

**3.5.2 O transporte de pacotes sobre ADSL**

O transporte de pacotes sobre ADSL (Asymmetric Digital Subscriber Line) envolve uma série de protocolos e dispositivos para permitir a conexão de milhões de assinantes domésticos à Internet através do circuito terminal de telefone. Aqui está uma descrição detalhada do processo:

1. **Camada Física ADSL:**
   * A ADSL utiliza um esquema de modulação digital baseado em multiplexação por divisão de frequência ortogonal para transmitir dados sobre o circuito terminal de telefone.
2. **Protocolos e Dispositivos Envolvidos:**
   * Na extremidade do cliente, um computador envia pacotes IP para um modem DSL através de uma camada de enlace, como Ethernet padrão.
   * O modem DSL envia os pacotes IP para o DSLAM (DSL Access Multiplexer), localizado na estação local da companhia telefônica, usando protocolos específicos.
3. **Protocolos Utilizados na ADSL:**
   * Entre a camada física ADSL e o PPP está o protocolo ATM (Asynchronous Transfer Mode) e o AAL5 (ATM Adaptation Layer 5).
   * O ATM é uma camada de enlace baseada na transmissão de células de informação de tamanho fixo, enquanto o AAL5 é uma camada de adaptação ATM utilizada para mapear os dados em células ATM.
   * O PPP (Point-to-Point Protocol) é utilizado para estabelecer e configurar o enlace e transportar pacotes IP. O padrão PPPoA (PPP over ATM) descreve como trabalhar com quadros PPP e AAL5 sobre a infraestrutura ATM.
4. **Funcionamento do PPPoA:**
   * No PPPoA, apenas os campos de protocolo e carga útil do PPP são colocados na carga útil do AAL5, pois o ATM já fornece o enquadramento necessário.
   * O campo de protocolo indica ao DSLAM se a carga útil é um pacote IP ou de outro protocolo, como LCP (Link Control Protocol).
   * Dentro do quadro AAL5, o enquadramento PPP não é necessário, pois o ATM e o AAL5 já fornecem o enquadramento. Além disso, o CRC (Cyclic Redundancy Check) do PPP não é necessário, pois o AAL5 já inclui seu próprio CRC para detecção de erros.
5. **Detecção e Correção de Erros na ADSL:**
   * A ADSL utiliza um esquema de detecção e correção de erros mais sofisticado do que em outros sistemas, incluindo um código de Reed-Solomon para correção de erro e um CRC de 1 byte para detecção de erros restantes.
   * Esse esquema é essencial para lidar com o ruído associado ao canal ADSL e garantir a integridade dos dados transmitidos.

Esse processo garante uma conexão estável e confiável dos assinantes domésticos à Internet através da infraestrutura ADSL, permitindo o transporte eficiente de pacotes IP sobre o circuito terminal de telefone.