semana 9- aula 01
Protocolos e Camadas
Camada de Enlace: Controle de Acesso ao Meio e Detecção de Erros
Código da aula: [SIS]ANO1C2B2S9A1

Objetivos da Aula:

- ❖ Compreender o controle de acesso ao meio e a detecção de erros da camada de enlace com uso de hubs.
- ❖ Conhecer técnicas de computação e gerenciamento de dados para soluções em nuvem, parametrizando aplicações e dimensionando de acordo com as necessidades do negócio;
- ❖ Identificar e analisar problemas;

Exposição:

A Camada de Enlace (Data Link Layer), a segunda camada do Modelo OSI, tem como principal objetivo fornecer um meio confiável para a transferência de dados entre dois nós diretamente conectados em uma rede física. Ela atua como uma ponte entre a Camada Física (que lida com a transmissão bruta de bits) e a Camada de Rede (responsável pelo roteamento entre diferentes redes).

Pense na Camada de Enlace como o "carteiro local" da sua rede. Ela garante que a correspondência (seus dados) seja entregue de forma segura e eficiente entre vizinhos (dispositivos na mesma rede).

As principais funções da Camada de Enlace incluem:

- Enquadramento (Framing): Recebe os pacotes da Camada de Rede e os encapsula em unidades menores chamadas quadros (frames). Adiciona cabeçalhos e trailers aos dados, contendo informações de controle como endereços físicos (MAC), detecção de erros e controle de fluxo.
- Endereçamento Físico (MAC Addressing): Utiliza endereços MAC (Media Access Control) para identificar unicamente os dispositivos dentro da mesma rede local. Esses endereços são físicos, gravados na placa de interface de rede (NIC) de cada dispositivo.
- Controle de Acesso ao Meio (Media Access Control - MAC): Define as regras para que múltiplos dispositivos compartilhem o mesmo meio físico de transmissão (como um cabo Ethernet ou uma frequência Wi-Fi) sem que ocorram colisões e garantindo um acesso justo.
- Detecção e, em alguns casos, Correção de Erros: Implementa mecanismos para detectar erros que possam ocorrer durante a transmissão na Camada Física (ruídos, interferências). Alguns protocolos também possuem a capacidade de corrigir esses erros ou solicitar a retransmissão dos dados corrompidos.

- Controle de Fluxo: Regula a taxa de transmissão de dados entre dois nós para evitar que um transmissor rápido sobrecarregue um receptor mais lento, garantindo que os dados sejam processados de forma adequada.

Exemplos de protocolos e tecnologias da Camada de Enlace:

- Ethernet (IEEE 802.3): O protocolo mais amplamente utilizado em redes locais (LANs) com fio. Define o formato dos quadros, o método de acesso ao meio (CSMA/CD em versões mais antigas, comutação em redes modernas) e as características físicas da conexão.
- Wi-Fi (IEEE 802.11): O padrão para redes locais sem fio (WLANs). Define o formato dos quadros, o método de acesso ao meio (CSMA/CA) e os mecanismos de segurança.
- PPP (Point-to-Point Protocol): Um protocolo usado para estabelecer conexões diretas (ponto a ponto) entre dois nós, como em conexões discadas, DSL ou conexões WAN seriais. Fornece autenticação, criptografia e compressão.
- HDLC (High-Level Data Link Control): Um protocolo de enlace de dados síncrono, orientado a bits, utilizado em redes WAN.
- Frame Relay: Uma tecnologia de comutação de pacotes utilizada em redes WAN, operando nas camadas física e de enlace.
- ATM (Asynchronous Transfer Mode): Outra tecnologia de comutação de pacotes para redes WAN, caracterizada por células de tamanho fixo.

Em resumo, a Camada de Enlace é crucial para garantir uma comunicação confiável e organizada dentro de uma mesma rede física, preparando os dados para serem roteados para seus destinos finais pela Camada de Rede.

No contexto da Camada de Enlace do modelo OSI, a Detecção de Erros refere-se aos mecanismos implementados para identificar se os dados transmitidos através do meio físico foram alterados ou corrompidos durante o processo. Como a Camada Física é suscetível a ruídos, interferências e outros problemas que podem introduzir erros nos sinais elétricos, ópticos ou de rádio, a Camada de Enlace incorpora técnicas para verificar a integridade dos dados recebidos.

O objetivo principal da detecção de erros na Camada de Enlace é:

- Garantir a confiabilidade da comunicação local: Ao identificar quadros corrompidos, a camada pode descartá-los, evitando que dados errôneos sejam passados para as camadas superiores (especialmente a Camada de Rede).
- Solicitar retransmissão (em alguns casos): Alguns protocolos da Camada de Enlace, ao detectarem um erro, podem solicitar automaticamente a retransmissão do quadro corrompido pelo transmissor, garantindo a entrega correta dos dados.

Como a Detecção de Erros funciona na Camada de Enlace:

A detecção de erros geralmente envolve a adição de informações redundantes (bits extras) aos quadros de dados antes da transmissão. O dispositivo receptor realiza cálculos sobre os dados recebidos e as informações redundantes para verificar se houve alguma alteração. Se os cálculos não coincidirem, um erro é detectado.

Técnicas comuns de Detecção de Erros na Camada de Enlace:

- Bit de Paridade: Um bit adicional (paridade par ou ímpar) é adicionado a cada unidade de dados. O número total de bits '1' (incluindo o bit de paridade) deve ser par ou ímpar, dependendo do esquema de paridade utilizado. Se o receptor verificar uma paridade incorreta, um erro é detectado. É uma técnica simples, mas detecta apenas um número ímpar de erros.
- Checksum: Um valor numérico (checksum) é calculado com base nos dados transmitidos. Esse valor é incluído no quadro. O receptor realiza o mesmo cálculo nos dados recebidos e compara o resultado com o checksum recebido. Se os valores forem diferentes, um erro é detectado. O checksum é mais eficaz que o bit de paridade na detecção de múltiplos erros.
- CRC (Cyclic Redundancy Check): Uma técnica mais sofisticada e amplamente utilizada. Um polinômio gerador é usado para calcular um código de redundância cíclica (CRC) com base nos dados. Esse código é anexado ao quadro. O receptor realiza uma divisão polinomial nos dados recebidos usando o mesmo polinômio gerador. Se o resto da divisão for diferente de zero, um erro é detectado. O CRC é muito eficaz na detecção de uma grande variedade de erros, incluindo erros de burst (sequências de bits errados).

Exemplos no contexto dos protocolos da Camada de Enlace:

- Ethernet: Utiliza o CRC (normalmente CRC-32) para detecção de erros nos seus quadros.
- Wi-Fi: Também emprega o CRC para garantir a integridade dos dados transmitidos sem fio.
- HDLC e PPP: Podem utilizar o CRC como um dos seus mecanismos de detecção de erros.

É importante notar que a detecção de erros na Camada de Enlace não garante a ausência total de erros (principalmente em casos de múltiplos erros que se cancelam). No entanto, ela reduz significativamente a probabilidade de dados corrompidos serem passados para as camadas superiores, contribuindo para uma comunicação mais confiável dentro da rede local. A correção de erros, quando implementada na Camada de Enlace, geralmente envolve a retransmissão dos quadros detectados como errôneos.

semana 9- aula 02
Protocolos e Camadas
Camada de Enlace: Controle de Acesso ao Meio e Detecção de Erros
Código da aula: [SIS]ANO1C2B2S9A2

Objetivos da Aula:

- Compreender o controle de acesso ao meio e a detecção de erros da camada de enlace com uso de switches.
- Conhecer técnicas de computação e gerenciamento de dados para soluções em nuvem, parametrizando aplicações e dimensionando de acordo com as necessidades do negócio;
- Identificar e analisar problemas;

Exposição:

O Endereçamento MAC, também conhecido como endereçamento físico, é um identificador único de 48 bits (geralmente representado em formato hexadecimal) atribuído à placa de interface de rede (NIC) de um dispositivo. Ele opera na subcamada MAC (Media Access Control) da Camada de Enlace do Modelo OSI e tem como principal função identificar um dispositivo especificamente dentro de uma mesma rede local (LAN).

Pense no endereço MAC como o número de série gravado na sua placa de rede. Ele é único para cada interface de rede no mundo, teoricamente falando, pois é atribuído pelo fabricante do hardware.

As principais características do Endereçamento MAC são:

- Unicidade: Cada placa de rede fabricada deve ter um endereço MAC globalmente único. Isso evita conflitos de identificação dentro de uma rede local.
- Físico: O endereço MAC é gravado diretamente no hardware da placa de rede (na memória ROM ou outro tipo de memória não volátil) e, portanto, é considerado um endereço físico.
- Local: O endereçamento MAC é relevante apenas dentro do domínio de uma rede local. Roteadores utilizam endereços IP (da Camada de Rede) para encaminhar dados entre diferentes redes.
- Formato: Geralmente é representado por seis grupos de dois dígitos hexadecimais, separados por dois pontos, hífens ou sem separação (ex: `00:1A:2B:3C:4D:5E` ou `00-1A-2B-3C-4D-5E` ou `001A2B3C4D5E`).
- Estrutura: Os primeiros 24 bits do endereço MAC (conhecidos como OUI - Organizationally Unique Identifier) identificam o fabricante da placa de rede.

Os últimos 24 bits são atribuídos pelo fabricante para identificar unicamente a interface.

Em resumo, o Endereçamento MAC ou físico é a identidade de hardware de uma placa de rede, essencial para a comunicação e identificação de dispositivos dentro de uma rede local. Ele permite que switches e outros dispositivos da Camada de Enlace encaminhem corretamente os quadros de dados para o destino correto dentro do segmento de rede.

Em redes que utilizam switches, a detecção de erros ocorre principalmente no nível da Camada de Enlace, dentro do próprio switch, para garantir a integridade dos quadros que ele recebe e encaminha. Os switches implementam mecanismos de detecção de erros de forma semelhante a outros dispositivos da Camada de Enlace, focando em cada quadro individualmente.

Como a detecção de erros funciona em switches:

1. Recepção do Quadro: Quando um switch recebe um quadro em uma de suas portas, ele examina o quadro completo antes de tomar uma decisão sobre o seu encaminhamento (no modo de comutação "armazenar e encaminhar" - *store-and-forward*).

2. Verificação do Campo de Verificação de Sequência de Quadro (FCS - Frame Check Sequence): Cada quadro Ethernet (o protocolo mais comum em redes com switches) possui um campo chamado FCS no seu final (o *trailer* do quadro). Este campo contém um valor de CRC (Cyclic Redundancy Check).

3. Cálculo do CRC: O switch realiza o mesmo cálculo de CRC sobre os dados recebidos (os bytes do quadro entre os campos de endereço de destino e origem até o campo de dados) utilizando o mesmo polinômio gerador que o dispositivo transmissor utilizou para gerar o CRC original.

4. Comparação dos Valores de CRC: O valor de CRC calculado pelo switch é então comparado com o valor de CRC contido no campo FCS do quadro recebido.

5. Detecção de Erro:

    - Se os valores de CRC coincidirem: Isso indica que, muito provavelmente, os dados não foram corrompidos durante a transmissão entre o dispositivo de origem e o switch. O switch então prossegue com o processo de encaminhamento do quadro para a porta de destino apropriada.

    - Se os valores de CRC não coincidirem: Isso significa que um erro (ou múltiplos erros) ocorreu durante a transmissão. Nesse caso, o switch descarta o quadro defeituoso.

Importância da Detecção de Erros em Switches:

- Prevenção da Propagação de Dados Corrompidos: Ao descartar quadros com erros, os switches impedem que dados defeituosos se propaguem pela rede, evitando problemas de comunicação e inconsistências em outros dispositivos.
- Melhoria da Confiabilidade da Rede Local: A detecção de erros contribui para uma rede local mais confiável, garantindo que apenas dados íntegros sejam entregues aos dispositivos de destino.
- Diagnóstico de Problemas na Camada Física: Um alto número de quadros descartados devido a erros de CRC em uma porta específica de um switch pode indicar problemas na Camada Física, como cabos defeituosos, interferência eletromagnética ou problemas nas placas de rede dos dispositivos conectados a essa porta.

Observação: A maioria dos switches modernos opera no modo "armazenar e encaminhar" para garantir a integridade dos dados. Alguns switches mais antigos ou com configurações específicas podem operar em modo "corte direto" (*cut-through*), onde o encaminhamento do quadro começa antes que ele seja completamente recebido. Nesse modo, a detecção de erros pode ser limitada ou inexistente antes do início do encaminhamento, o que pode levar à propagação de quadros defeituosos. No entanto, o modo "armazenar e encaminhar" é o padrão para garantir a confiabilidade.

semana 9- aula 03
Protocolos e Camadas
Camada de Enlace: Controle de Acesso ao Meio e Detecção de Erros
Código da aula: [SIS]ANO1C2B2S9A3

Objetivos da Aula:

- Compreender as técnicas utilizadas na detecção de erros da camada de enlace.
- Conhecer técnicas de computação e gerenciamento de dados para soluções em nuvem, parametrizando aplicações e dimensionando de acordo com as necessidades do negócio;
- Identificar e analisar problemas;

Exposição:

Checksums são um tipo de técnica de detecção de erros utilizada em redes de computadores e armazenamento de dados para verificar a integridade dos dados transmitidos ou armazenados. Em essência, um checksum é um valor numérico calculado a partir de um bloco de dados. Esse valor é transmitido ou armazenado junto com os dados. O receptor (ou o sistema que lê os dados armazenados) realiza o mesmo cálculo sobre os dados recebidos/lidos e compara o resultado com o checksum recebido/armazenado.

Se os checksums coincidirem, é altamente provável que os dados não tenham sido alterados ou corrompidos durante a transmissão ou armazenamento.

Se os checksums não coincidirem, isso indica que houve uma alteração nos dados.

Como funciona o cálculo de um Checksum (de forma simplificada):

Existem diversas maneiras de calcular um checksum, mas uma das mais simples envolve os seguintes passos:

1. Dividir os dados: O bloco de dados é dividido em segmentos de tamanho fixo (por exemplo, bytes ou palavras de 16 bits).
2. Somar os segmentos: Esses segmentos são somados aritmeticamente.
3. Complemento de um (opcional): Em algumas implementações, o resultado da soma é complementado de um (todos os bits são invertidos).
4. O resultado é o Checksum: O valor final da soma (possivelmente complementado) é o checksum.

Exemplo Simplificado (usando bytes):

Suponha que os dados a serem transmitidos sejam os bytes: `0x45`, `0x6E`, `0x64`.

1. Soma: `0x45 + 0x6E + 0x64 = 0x157`

2. Truncamento (se necessário): Se o checksum for limitado a um byte, o overflow seria descartado, resultando em `0x57`.
3. Checksum: O checksum a ser transmitido junto com os dados seria `0x57`.

O receptor receberia os bytes `0x45`, `0x6E`, `0x64` e o checksum `0x57`. Ele realizaria a mesma soma (`0x45 + 0x6E + 0x64 = 0x157`) e, aplicando o mesmo truncamento (se necessário), obteria `0x57`. Como o checksum calculado coincide com o checksum recebido, assume-se que os dados foram transmitidos sem erros.

Características dos Checksums:

- Simples de implementar: Em geral, os algoritmos de checksum são relativamente fáceis de implementar em hardware e software.
- Rápido de calcular: O cálculo do checksum é geralmente rápido, adicionando pouca sobrecarga ao processo de transmissão ou armazenamento.
- Detecção de erros: São eficazes na detecção de muitos tipos de erros, especialmente erros aleatórios de um único bit e alguns erros de múltiplos bits.
- Limitações: Checksums mais simples podem não detectar todos os tipos de erros, especialmente se ocorrerem múltiplos erros que se "cancelam" na soma. Técnicas mais robustas como CRC (Cyclic Redundancy Check) oferecem uma maior capacidade de detecção de erros.

Onde os Checksums são utilizados:

- Protocolos de rede: Protocolos como TCP (na camada de transporte) e IP (na camada de rede) utilizam checksums para verificar a integridade dos seus cabeçalhos e, em alguns casos, dos dados.
- Armazenamento de dados: Alguns sistemas de arquivos e softwares de backup utilizam checksums para verificar se os arquivos foram corrompidos durante o armazenamento ou a transferência.
- Download de arquivos: Muitas vezes, ao baixar arquivos da internet, um checksum (geralmente em formatos como MD5 ou SHA) é fornecido para que o usuário possa verificar se o arquivo baixado está completo e não foi corrompido.

Em resumo, checksums são valores numéricos derivados de dados que servem como uma "assinatura" para verificar a integridade desses dados após a transmissão ou armazenamento. Eles oferecem uma maneira rápida e relativamente simples de detectar muitos tipos de erros, embora possam ter limitações em comparação com técnicas mais avançadas de detecção de erros.

Análise da diferenças entre Checksum e CRC

Eficiência na Detecção de Erros:

- Checksum: É um método simples que basicamente soma os valores dos dados transmitidos e usa o resultado (com algumas variações, como complemento de um) como um valor de verificação. Ele é eficiente para detectar erros simples, como a alteração de um único bit ou um pequeno número de bits adjacentes. No entanto, o checksum pode falhar em detectar erros mais complexos, como a troca de ordem de bits ou múltiplos erros que se cancelam na soma.

- CRC (Cyclic Redundancy Check): Utiliza a divisão polinomial em um campo finito para gerar um código de redundância. Essa abordagem torna o CRC significativamente mais eficiente na detecção de uma ampla gama de erros, incluindo erros únicos, múltiplos erros isolados, erros em burst (sequências de bits errados) e até mesmo alguns padrões de erros mais complexos. A eficiência do CRC depende do polinômio gerador escolhido; polinômios mais longos geralmente oferecem maior poder de detecção.

Complexidade de Implementação:

- Checksum: A implementação do checksum é relativamente simples. Envolve operações básicas de adição e, possivelmente, complementação. Isso torna o checksum fácil de entender e implementar em hardware e software, exigindo poucos recursos computacionais.

- CRC: A implementação do CRC é mais complexa. Envolve operações de divisão polinomial, que podem ser implementadas usando registradores de deslocamento com feedback XOR (LFSRs) em hardware ou por meio de algoritmos mais elaborados em software. Embora a lógica subjacente não seja excessivamente difícil, a implementação eficiente, especialmente em hardware de alta velocidade, requer um conhecimento mais aprofundado.

Velocidade de Processamento:

- Checksum: Devido à sua simplicidade, o cálculo do checksum é geralmente muito rápido. As operações de adição são rápidas e podem ser facilmente otimizadas. Isso torna o checksum adequado para aplicações onde a velocidade é crucial e a taxa de erro esperada é baixa.

- CRC: O processamento do CRC é mais lento que o checksum, especialmente quando implementado em software. As operações de divisão polinomial exigem mais ciclos de processamento. No entanto, implementações em hardware podem alcançar velocidades muito altas, tornando o CRC viável para aplicações de alta velocidade, como redes de comunicação. A velocidade do CRC depende do tamanho do polinômio gerador e da eficiência da implementação.

Aplicações Ideais:

- Checksum:
    - Aplicações com baixa probabilidade de erros: Onde erros complexos são raros e a velocidade é primordial.
    - Verificações rápidas e simples: Em cenários onde uma camada adicional de detecção de erros, mesmo que básica, é desejada com baixo overhead.
    - Alguns protocolos de rede mais antigos ou mais simples: Onde a complexidade era uma grande preocupação.
    - Verificação de integridade de arquivos pequenos: Onde a velocidade da verificação é mais importante que a robustez contra erros complexos.
- CRC:
    - Comunicação de dados: Amplamente utilizado em protocolos de rede (Ethernet, Wi-Fi), armazenamento de dados (HDDs, SSDs), compressão de dados (ZIP, RAR) e transmissão digital (DVD, Blu-ray) devido à sua alta capacidade de detecção de erros.
    - Aplicações onde a integridade dos dados é crucial: Onde a detecção de uma ampla gama de erros é fundamental para garantir a confiabilidade.
    - Sistemas com maior probabilidade de ruído ou interferência: Onde erros em burst são mais comuns.