UniFil
EDUCAÇÃO A DISTÂNCIA

Unidade 2

Meios Físicos e Camada de Enlace

Abertura

Apresentação

Esta unidade elenca alguns meios físicos e sua tecnologia e funcionamento, bem como discutimos sobre a camada de enlace, que é de certa forma, acoplada ao meio físico. Como assim físicos, você pode estar pensando, mas é exatamente isso, os cabos e seus materiais físicos, bem como o ar e as ondas por onde surfam os dados. A camada física é a mais baixa da hierarquia, e define praticamente todo o funcionamento e confiabilidade também.

Aprenderemos também, os modelos e padrões que consistem numa parte fundamental da disciplina, formalizando determinados tópicos para que as redes funcionem como a vemos e usamos. Com isto, faremos um breve passeio pelas camadas dos modelos, e a proposta de cada uma.

Objetivos

- Conhecer os objetos de estudo de Redes de Computadores;
- Entender as formas de classificação das redes;
- Compreender a divisão entre camadas e as responsabilidades de cada camada.

Conteúdo programático

Aula 01 – Camada Física

Aula 02 – Camada de Enlace

A Camada Física

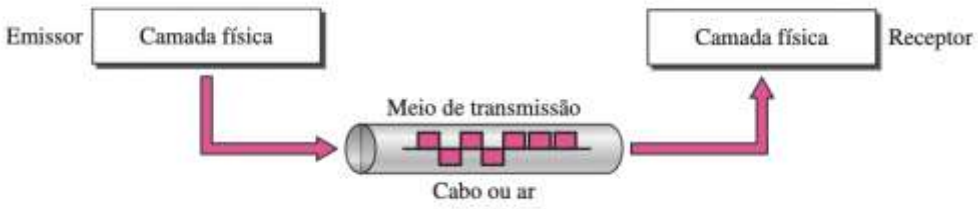
A primeira coisa que devemos entender é que o computador e as redes de computadores apenas entendem 0 e 1, o que chamamos de ***bit*** (do inglês, *binary digit*). A música que escutamos, o vídeo a que assistimos ou a foto que damos um "*like*", tudo o que conseguimos armazenar ou transmitir é um conjunto de zeros e uma armazenada no computador. Basicamente, um *bit* consegue assumir apenas dois estados discretos, e com isto, nós conseguimos mapear para usarmos duas voltagens distintas, ou duas intensidades de luz diferentes, entre outros.

Dentro do computador, todos os dados são processados e armazenados como binários para representar e interpretar letras, números e caracteres especiais com bits. O código ASCII (*American Standard Code for Information Interchange*) é comumente utilizado, e faz a representação usando oito *bits*. Cada oito *bits*, nós chamamos de byte. Assim, 00000000 é um byte e de acordo com a tabela ASCII representa NUL. Abaixo alguns exemplos de caracteres representados na codificação ASCII:

- A - 01000001
- a - 01100001
- 9 = 00111001
- # = 00100011

Precisamos entender que os dados que iremos transmitir, geralmente estão em um digital, e talvez sofra uma alteração para uma forma analógica que, por sua vez, no receptor, será transformada em digital novamente. Depois que os dados são transformados em uma série de bits, eles devem ser convertidos em sinais que podem ser enviados até seu destino. Nos meios de transmissão, existem três métodos comuns usados em redes: sinais elétricos, sinais ópticos e sinais sem fio.

Figura 1: Meios de transmissão e a camada física



Fonte: FOROUZAN, 2009.

No caso da sala de aula, o meio de transmissão é o ar, pois utilizamos a voz para nos comunicarmos e esta, por sua vez, se propaga através das ondas sonoras, um meio sem fio. A informação também está codificada na língua portuguesa, que é como se fosse a nossa tabela ASCII. Ou seja, o professor envia a informação para os alunos através da voz que se propaga no ar e como falamos a mesma língua, vocês conseguem receber a informação e entender. Se alguém na sala de aula não entendesse o português, teríamos dificuldade na comunicação.

O início da comunicação em longa distância foi usando sinais elétricos e dando início a era da informação através dos telégrafos, onde tinha uma comunicação lenta e dependente de fios. Para enviar ou receber a informação, era necessário o conhecimento do código Morse, muito utilizado ainda hoje por radioamadores. Anos depois, nós conseguimos enviar a voz humana com o advento do telefone, entretanto, no início as linhas eram ruidosas e a comunicação não era confiável devido à baixa qualidade dos fios. Tivemos também o envio de informações com comunicação sem fio através da transmissão de sinais de alta frequência por Hertz em meados de 1895.

Podemos dividir os meios de transmissão em duas categorias: meios guiados e meios não guiados. Entre os meios de transmissão guiados temos os cabos de modo geral: par trançado, coaxial e fibra óptica. O meio de transmissão não guiado é sem fios.

Meios de Transmissão Guiados

Os meios de transmissão guiados trafegam o sinal através de um condutor físico que liga as pontas entre o emissor e receptor, e é limitado pelo meio físico. Geralmente são usados metais, mas transportam sinais na forma de luz também. Nas próximas seções discutiremos sobre: **par trançado**, **coaxial** e **fibra óptica**.

Cabo par trançado

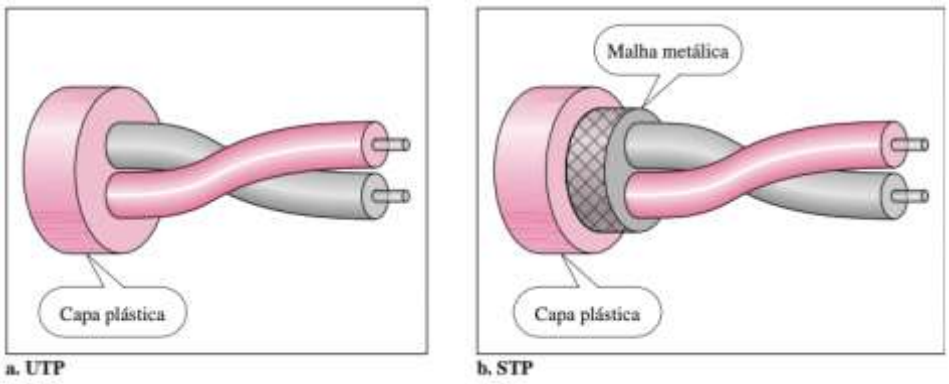
O cabo par trançado, geralmente feito por dois fios de cobre revestido por material isolante são trançados, por isto nome de "**par trançado**". Um dos fios transporta os sinais, e o outro é uma referência, assim, o receptor calcula a diferença de potencial entre os fios e determina a amplitude do sinal. Além disso, a forma trançada auxilia na redução de ruídos, e não teria o mesmo efeito se os fios estivessem apenas em paralelo.

Figura 2: Meios de transmissão e a camada física

Fonte: FOROUZAN, 2009.

Para auxiliar na redução de ruídos, temos duas formas de cabo de par trançado: **blindado** (STP - do inglês, *Shield Twisted Pair*) e **não blindado** (UTP - do inglês, *Unshield Twisted Pair*). Na grande maioria dos casos, vamos encontrar cabos UTP, e em raras ocasiões, STP.

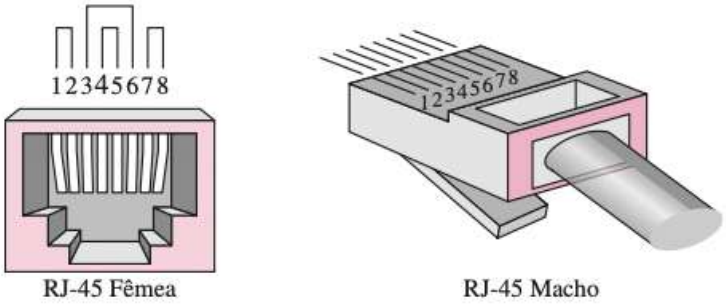
Figura 3: Cabos UTP e STP



Fonte: FOROUZAN, 2009.

Os conectores mais utilizados e comumente encontrados são o RJ-11 e RJ-45. **RJ** é ***R**egistered **J**ack*, foi introduzido pela Bell System e criou uma padronização para muitos conectores modulares. O RJ-11 é utilizado no sistema telefônico e possui 6 Pinos e 4 Conectores, o padrão 6P4C. Já o RJ-45 é chamado assim de maneira "errada" pois segue o padrão ANSI/TIA-968-A e não o esquema proposto originalmente, e é utilizado para Ethernet num padrão 8P8C.

Figura 4: Conector Fêmea e Macho RJ-45



Fonte: FOROUZAN, 2009.

Os cabos de par trançado são usados em linhas telefônicas para a transmissão de voz e de dados, encontrados no chamado ***loop local*** (último trecho que conecta os assinantes ao prédio da central telefônica) e geralmente consistem em cabos de par trançado não blindados. A tecnologia DSL também usa esse tipo de ligação, e também as redes locais. Quando falamos de redes locais, é o escritório, sua casa, sua universidade. Basta olhar em volta, e provavelmente, você encontrará alguns cabos de cor azul passando em alguma canaleta.

Os cabos seguem uma categoria, definida pela EIA (*Eletronic Industries Association)*, que fundamentalmente utilizam a qualidade do cabo como determinante. A Tabela 1 abaixo mostra estas categorias.

Tabela 1: Categorias de cabos de par trançado

Categoria	*Especificação*	*Taxa de Dados (Mbps)*	*Uso*
1	Cabo de par trançado usado em telefonia	< 0,1	Telefonia
2	Cabo de par trançado não blindado usado originalmente em linhas T	2	Linhas T-1
3	CAT 2 aperfeiçoado para uso em redes locais LANs	10	LANs
4	CAT 3 aperfeiçoado para uso em redes Token Ring	20	LANs
5	O fio do cabo é, normalmente, AWG 24 com um invólucro e revestimento externo	100	LANs
5E	Extensão da categoria 5 que inclui recursos adicionais para reduzir interferências eletromagnéticas e linha cruzada	125	LANs
6	Uma nova categoria com componentes casados provenientes do mesmo fabricante. O cabo tem de ser testado à taxa de dados de 200 Mbps.	200	LANs
7	Algumas vezes, denominado SSTP (*shielded screen twisted-pair*). Cada par é envolto individualmente por uma folha metálica helicoidal e depois por uma blindagem de folha metálica, além da cobertura externa. A blindagem diminui o efeito de linha cruzada e aumenta a taxa de dados.	600	LANs

Fonte: FOROUZAN, 2009.

No nosso dia a dia, o CAT5 e CAT5E são os mais comuns. Com o aumento da velocidade dos provedores, faz-se necessário uma infraestrutura melhor, logo, o custo dos equipamentos também aumenta um pouco. Em contrapartida, aumenta também a qualidade e confiabilidade.

Videoaula 1

Utilize o QR Code para assistir!

Agora, assista ao vídeo em que o professor aborda todo esse conteúdo introdutório e explica com mais detalhes o cabo de par trançado.

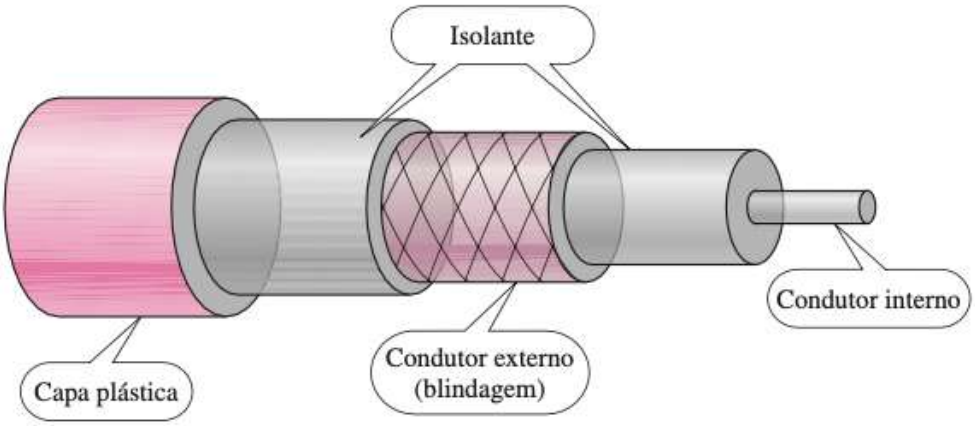
Cabo Coaxial

Antes da modernização das redes de telefonia, o coaxial foi amplamente utilizado, sendo capaz de transportar 10.000 sinais de voz no sistema analógico. Com a digitalização, era possível

atingir velocidade de até 600 Mbps. As redes de TV a cabo também utilizavam bastante, e ainda utilizam na extremidade da rede, nas nossas casas como conector final da rede. O cabo coaxial também foi muito usado nas redes LANs Ethernet tradicionais, dado a largura de banda, sendo largamente utilizado na Ethernet 10Base-2 e 10Base-5. Após a modernização com o aumento da tecnologia de fibra óptica e a redução dos custos, muito da infraestrutura hoje foi ou está sendo trocada para fibra óptica, mas temos operadoras que ainda, na casa do consumidor final utiliza o cabo coaxial.

O cabo coaxial tem uma estrutura de um fio condutor central revestido por um material isolante, seguido de uma blindagem externa feita de material condutor revestido por outra camada de material isolante, e por fim, uma capa plástica. A camada de blindagem externa é uma folha de metal que tem como objetivo a redução dos ruídos. Na Figura abaixo ilustra o cabo.

Figura 4: Cabo coaxial



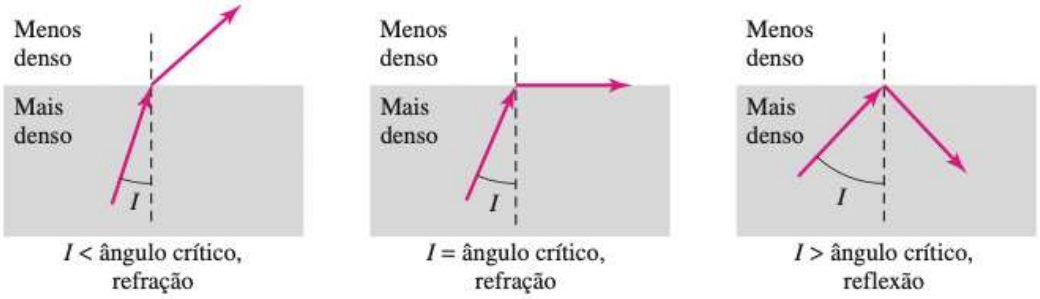
Fonte: FOROUZAN, 2009.

As faixas de frequência são mais altas que as do cabo de par trançado, permitindo enviar mais dados, em contrapartida, a atenuação do sinal no cabo coaxial é muito maior, necessitando o uso de repetidores.

Cabo Óptico

Enquanto os outros meios guiados se utilizam de um meio condutor, como um fio metálico, no caso de um cabo de fibra óptica, temos uma estrutura de vidro ou plástico, que permite enviarmos sinais na forma de luz. Utilizando dos conceitos de física e da natureza comportamental da luz, os conceitos de **refração** e **reflexão**, conseguimos trafegar informações em "forma de luz". Resumidamente, dado um raio de luz trafegando por um meio, ele altera sua direção de acordo com a diferença de densidade do meio trafegado e do meio que ele interceptou. Podemos ver o comportamento na Figura 5.

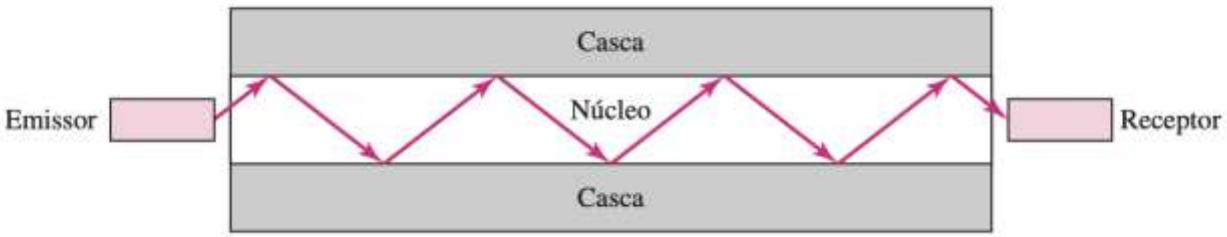
Figura 5: Desvio da luz



Fonte: FOROUZAN, 2009.

Como podemos observar na Figura 5, dependendo da densidade e do ângulo de incidência, o comportamento da luz se altera. Aproveitando-se dessa propriedade física, os cabos de fibra óptica são construídos de uma maneira que sempre ocorra a reflexão do sinal. Para isto, ele é feito de um núcleo de vidro, e mais recentemente de plástico, mais denso que o material que o reveste. Desta maneira, temos o comportamento desejado, conforme ilustra a Figura 6.

Figura 6: Luz percorrendo a fibra óptica

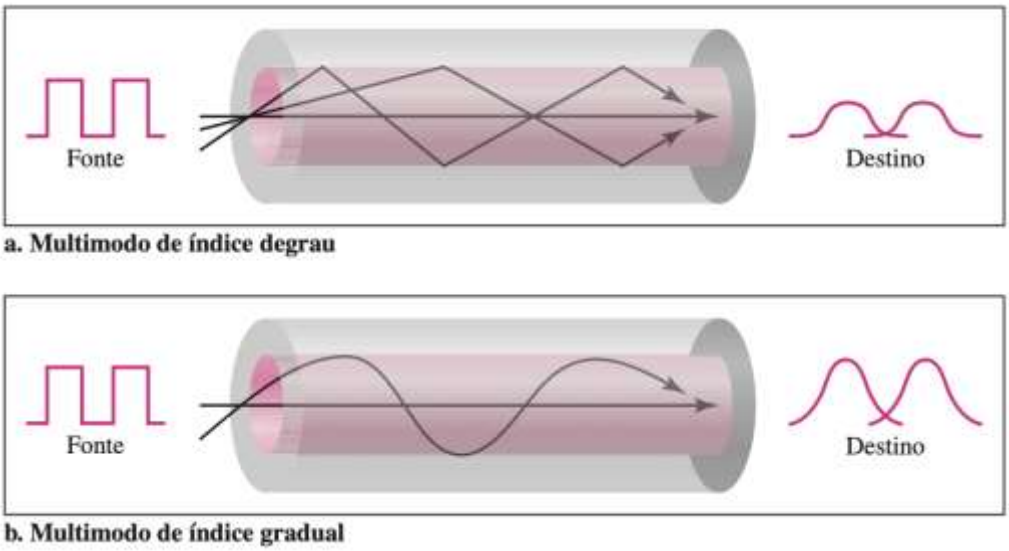


Fonte: FOROUZAN, 2009.

Dependendo do desempenho e da distância que queremos que o sinal percorra, temos dois modos para a propagação da luz nas fibras ópticas: **multimodo** e **monomodo.** Cada uma com características físicas diferentes, e no modo multimodo ainda temos duas formas: **índice degrau** e **índice gradual.**

Multimodo, como o nome sugere, é quando trabalhamos com múltiplos fluxos de uma fonte de luz que se deslocam ao longo do núcleo usando caminhos diferentes. Falamos que uma fibra óptica tem **índice degrau**, quando na interface há uma diferença bem grande entre as densidades. Já uma fibra com **índice gradual** refere-se a uma fibra com a densidade mais alta no centro e a diminuição gradativa para a borda.

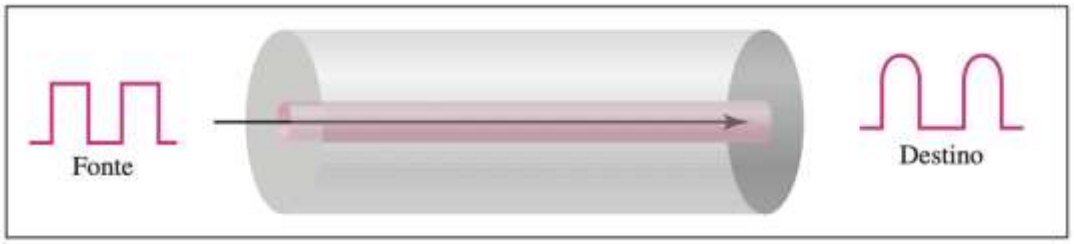
Figura 7: Fibra óptica Multimodo



Fonte: FOROUZAN, 2009.

A fibra **monomodo** é fabricada com um diâmetro muito menor que a da fibra multimodo e uma casca mais grossa. O espaço de transmissão é muito pequeno, fazendo com que o feixe de luz tenha um ângulo próximo de 90º, produzindo um comportamento como uma linha. Este modo é altamente eficaz, tendo uma perda menor de luz e percorrendo maiores distâncias sem repetidores de sinal.

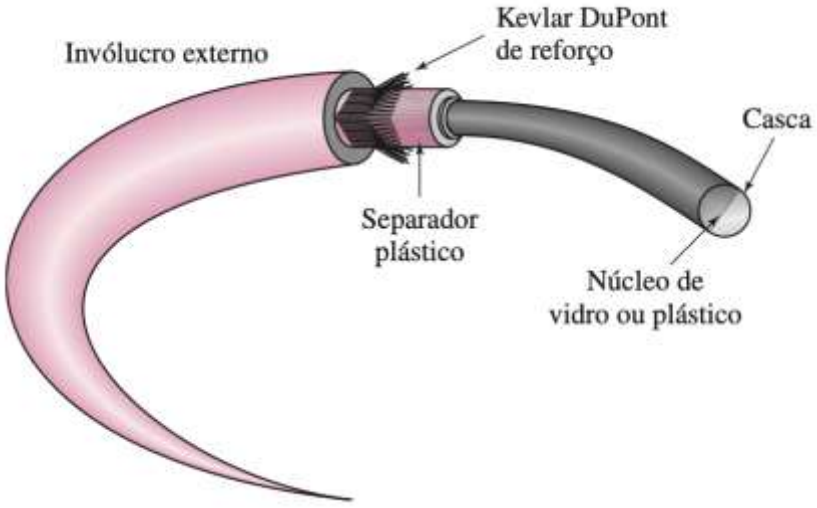
Figura 8: Fibra óptica Monomodo



Fonte: FOROUZAN, 2009.

A fibra óptica é formada por um invólucro externo em PVC ou Teflon. Temos uma malha de Kevlar para reforçar a estrutura do cabo, depois temos uma outra camada de revestimento plástico para proteger a fibra. A fibra óptica se encontra no centro do cabo, formada pela casca e pelo núcleo. As fibras ópticas são categorizadas pela razão entre o diâmetro de seus núcleos e o diâmetro de suas cascas, ambos expressos em micrômetros.

Figura 9: Composição da fibra óptica



Fonte: FOROUZAN, 2009.

Indicação de Vídeos

Para conhecer um pouco mais sobre os conectores de fibra óptica, este vídeo possui uma apresentação dos conectores e os modos.

Fiber Optic Connector Types Explained in Details. Disponível em: https://www.youtube.com/watch?v=4Ovqe3XjRqM. Acesso em: 19 mai. 2022.

Videoaula 2

Utilize o QR Code para assistir!

Agora, assista ao vídeo em que o professor explica o cabo coaxial e o cabo de fibra óptica.

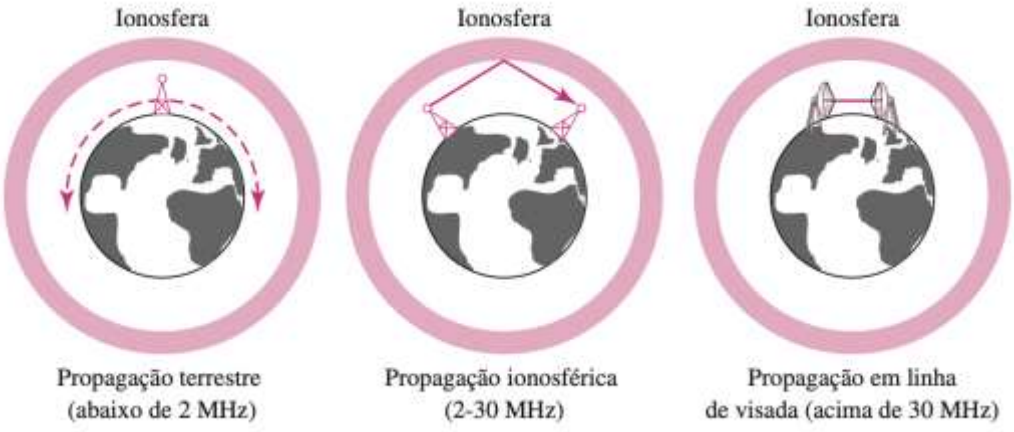
Meios de Transmissão Não Guiados

Ao contrário dos meios de transmissão guiados, os não guiados simplesmente não possuem um cabo ou qualquer outro dispositivo físico dessa natureza ligando o emissor e o receptor. Os sinais são transmitidos pelo espaço livre através das ondas eletromagnéticas, com isto, qualquer um que tenha um dispositivo com as capacidades de receber o sinal pode participar da comunicação. Discutimos brevemente sobre os espectros na aula 02 da unidade 01.

Existem três formas de propagar os dados nos meios não guiados (temos uma ideia do funcionamento ilustrado na Figura 10):

- **Propagação terrestre:** são sinais de baixa frequência que seguem a curvatura do planeta, e o alcance está diretamente relacionado à potência do sinal. As utilizações mais comuns são rádio AM, radionavegação de e localizadores de longo alcance.
- **Propagação ionosférica:** são sinais de alta frequência que são irradiados para cima atingindo a ionosfera onde são refletidas de volta para a Terra. Bastante aplicado em rádio AM e FM, comunicação de aeronaves e navios e TV VHF.
- **Propagação em linha de visada:** são utilizados sinais de frequência muito alta que são transmitidos em linha reta de uma antena unidirecional. É necessário mirar as antenas de maneira que fiquem uma voltada para a outra, e também altas o suficiente ou próximas o bastante para não serem afetadas pela curvatura da Terra. Utilizada nas TV UHF, telefones celulares e comunicação via satélites.

Figura 10: Métodos de propagação



Fonte: FOROUZAN, 2009.

Dentre os meios não guiados, encontramos as tecnologias de Wi-Fi, Bluetooth, infravermelho e outros. Todos se utilizam do espaço aberto para enviar e receber sinais. Com isto, finalizamos a Aula 03.

Videoaula 3

Utilize o QR Code para assistir!

Agora, assista ao vídeo em que o professor explica como funcionam os meios de transmissão não guiados e conclui a Aula 03.

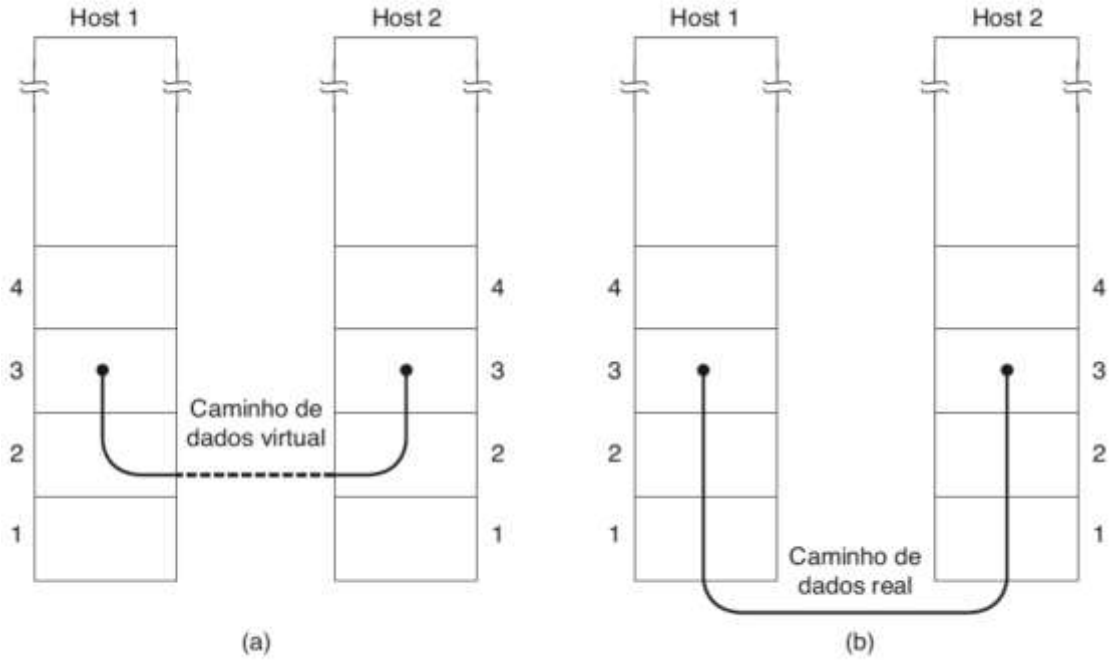
A Camada de Enlace

Nesta aula trataremos da camada de enlace, uma camada que separamos para melhor entendimento, mas é entrelaçada com a camada física. Quando falamos de comunicação entre dispositivos interligados em uma rede, os dispositivos devem ser capazes de transferir dados de um dispositivo a outro com precisão, garantindo que a informação enviada será a mesma recebida. Enquanto que na camada física definimos a interface, o meio que enviamos o sinal, a camada de enlace fica responsável por enquadrar (ou do inglês, *framing*), endereçamento, controle de fluxo, controle de erros e controle de acesso ao meio.

É nesta camada que enquadramos, ou seja, dividimos o fluxo de bits bruto da camada física em quadros, ou do inglês, *frames.* Além disso, adiciona-se um endereçamento para estabelecer o emissor e receptor. Também controla o fluxo, pois o emissor as vezes pode enviar muito mais informação que o receptor consiga tratar, e vice-versa. Adicionalmente, acrescenta confiabilidade com mecanismos de detecção de erros, e em alguns casos, até o tratamento dos frames.

O principal serviço é transferir dados da camada de rede da máquina de origem para a camada de rede da máquina de destino. Na camada de rede da máquina de origem há uma entidade, chamada processo, que entrega alguns bits à camada de enlace de dados para transmissão ao destino. A tarefa da camada de enlace de dados é transmitir bits à camada de rede dessa máquina de destino, conforme mostra a parte (a) da Figura 11. A transmissão propriamente dita é mostrada na parte (b), contudo, é mais fácil pensarmos em termos de dois processos na camada de enlace de dados, que se comunicam por intermédio de um protocolo de enlace de dados (por isso utilizaremos como referência sempre a parte (a).

Figura 11: Métodos de propagação



Fonte: TANENBAUM, 2011.

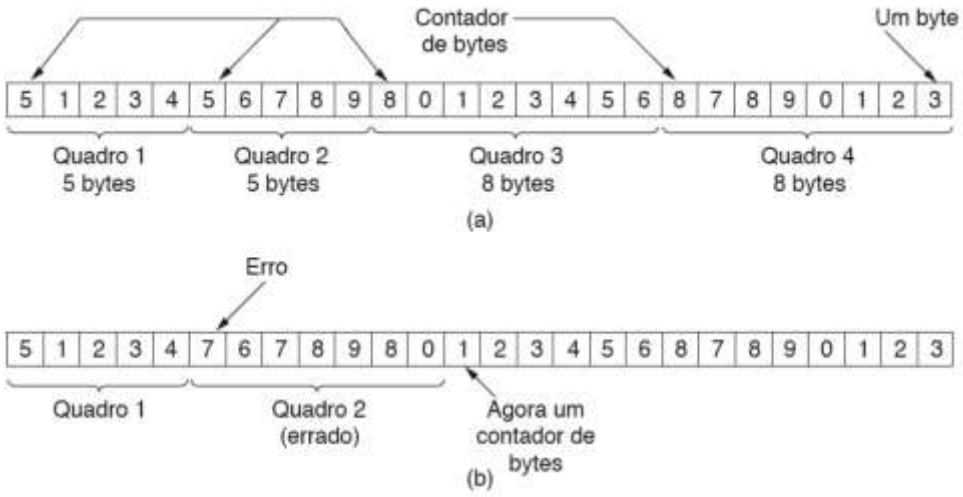
Framing (Enquadramento)

Para que o fluxo de bits seja entendido, a camada de enlace empacota os bits em ***frames***, ou **quadros**. É possível usarmos a analogia dos correios para um entendimento melhor. Quando queremos enviar uma carta ou algum item para alguém através dos correios, nós "empacotamos" em um envelope ou caixa contendo o item, o endereço do emissor e receptor. O que diferencia nossa encomenda das outras, ou seja, o que delimita uma da outra é o envelope. Assim, os correios sabem separar as mensagens (cartas e encomendas). É exatamente isto que o *framing* faz, ele cria esse envelope. Os frames geralmente são pequenos, apesar de ser possível empacotar uma mensagem longa inteira, isso não é feito pois impacta diretamente no controle de fluxo e de erros, podendo tornar esse processo ineficiente.

Podemos enquadrar os frames com tamanho **fixos** ou **variáveis**. No caso do tamanho fixo, o próprio tamanho é o delimitador do quadro. Com o tamanho **variável**, são necessários métodos eficientes para tal controle, os mais usados é o método orientado a caracteres e o método orientado a bit.

Um dos primeiros métodos orientados é a contagem de caracteres, em que enviamos um byte especial com o número de caracteres que fazem parte do quadro, com o byte especial incluso. Logo, se queremos enviar um quadro de 4 bytes, o byte especial terá o valor 5. Assim, o receptor recebe esse primeiro valor, e sabe quantos caracteres fazem parte deste quadro. Isto pode ser visto na Figura 12 (a).

Figura 12: Contagem de caracteres



Fonte: TANENBAUM, 2011.

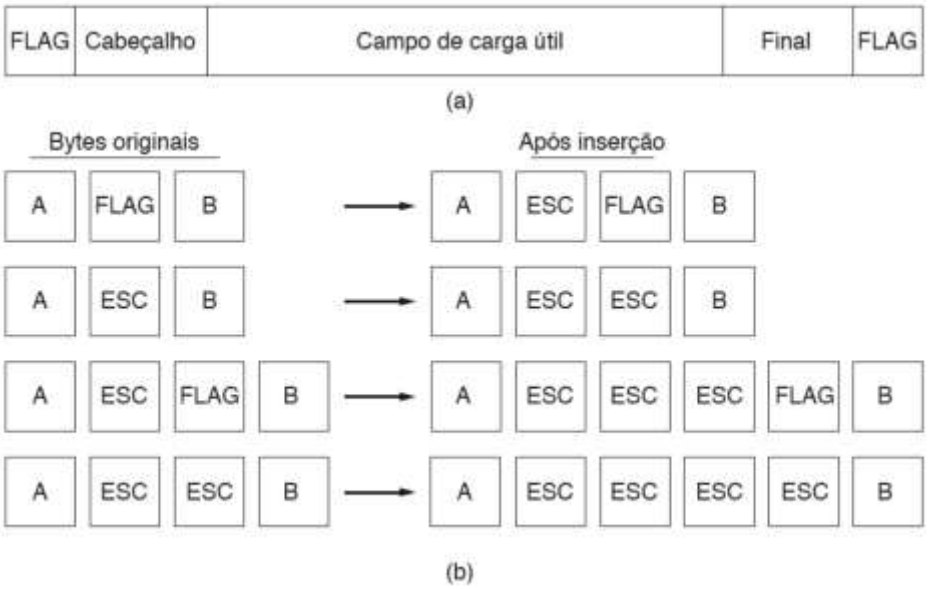
O problema com esse algoritmo é que a contagem pode ser adulterada por um erro de transmissão, conforme visto na parte (b) da Figura 12, fazendo com que seja perdido a capacidade de localizar o início do quadro seguinte. Mesmo que o *checksum* esteja incorreto, de modo que o destino identifique que o quadro está defeituoso, ele ainda não terá informações suficientes para saber onde começa o quadro seguinte

Enviar um quadro de volta à origem também não ajuda, pois, o destino não sabe quantos caracteres deverão ser ignorados para chegar ao início da retransmissão. Este é um dos motivos de o método de contagem de caracteres não ser utilizado, e pelo fato de a rede ter evoluído, e não apenas trocar informação textual.

Seguindo uma ideia similar, o *byte flag*, tenta contornar o problema de ressincronização após um erro, **fazendo cada quadro começar e terminar com bytes especiais**. Normalmente o mesmo *byte*, chamado de *byte flag*, é usado como delimitador de início e fim, conforme visto na parte (a) da Figura 13. Dois *bytes* de *flag* consecutivos indicam o fim de um quadro e o início de outro. Assim, se o receptor perder a sincronização ele poderá simplesmente procurar 2 *bytes* de *flag* para encontrar o final do quadro atual e o início do seguinte

Contudo ainda existe um problema, é bem possível que o byte de *flag* ocorra no meio dos dados transmitidos, interferindo no enquadramento. Nesta situação, uma solução é fazer com que o transmissor inclua um caractere de Escape Especial (ESC) imediatamente antes de cada *byte* de *flag* "acidental". A camada de enlace do receptor remove os *bytes* de escape antes de os entregar à camada de rede. Esta técnica é chamada de inserção de *bytes (byte stuffing).*

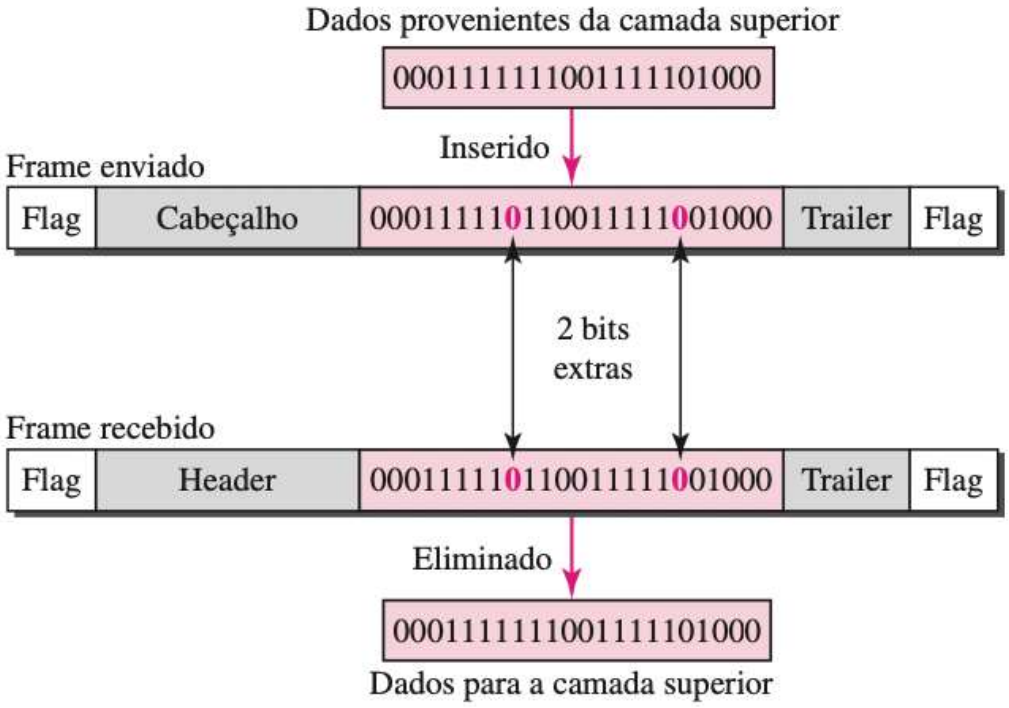
Figura 13: Byte flags



Fonte: TANENBAUM, 2011.

Quando o padrão da *flag* aparece no meio dos dados, precisamos informar o receptor que não se trata do aviso de fim de frame, e para isso, utilizamos a mesma estratégia do byte, mas a nível de bit, então chamamos de **bit-stuffing**. Toda vez que uma sequência 011111 for encontrada, adiciona-se um 0, independente do bit que venha a seguir. E no receptor, este 0 que foi acrescentado é removido, a Figura 14 ilustra esse processo.

Figura 14: Inserção e eliminação de bits



Fonte: FOROUZAN, 2009.

Isso significa que, se o padrão de *bits* parecido com o do *flag*, 01111110, aparecer no meio dos dados, ele mudará para 011111010 (bit inserido) e não será confundido com um *flag* pelo receptor. O *flag* real 01111110 não é inserido pelo emissor e será reconhecido como tal pelo receptor.

Videoaula 1

Utilize o QR Code para assistir!

Agora, assista ao vídeo em que o professor introduz a camada de enlace e discute o processo de framing.

Detecção e Correção de Erros

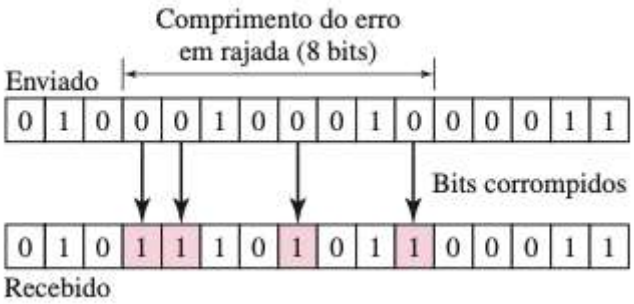
Já ressaltamos a importância da exatidão na transferência de dados em redes de computadores, e algumas aplicações toleram mais que outras, e neste contexto temos a camada de enlace auxiliando neste quesito. Detectar o erro é muito mais fácil que corrigir os

erros, pois na detecção de erros dizemos apenas se existe ou não erro. Não importa se o erro for de apenas um *bit* ou um bloco de *bits*, pois a mensagem está corrompida de qualquer maneira. Porém na correção de erros, temos que saber quantos *bits* estão corrompidos, e qual a localização deles na mensagem, para que assim, possamos corrigir a mensagem.

Dessa forma, já podemos imaginar como é uma tarefa complexa a correção de erros, pois o tamanho da mensagem e o número de erros são diretamente ligados e aumentam a complexidade. Em um *byte*, ou seja, 8 *bits*, se tivermos apenas um erro, temos 8 possibilidades, se forem dois erros em um *byte*, teremos 28 possibilidades. Imagine corrigir um erro em 1024 *bits*, ou 128 *bytes*, um valor pequeno ainda para uma mensagem, mas de alta complexidade para corrigir um erro.

Vimos que na camada física, estamos expostos às interferências do sinal que podem modificar os bits, fazendo com que um bit 0 passe a ser 1, e vice-versa. Isto é o que chamamos de **erro de bit**. Um **erro de rajada**, são vários bits corrompidos, como podemos observar na Figura 15.

Figura 15: Erro em rajada de comprimento 8



Fonte: FOROUZAN, 2009.

A chance de ocorrer um erro de *bit* é muito pequena, pois a duração de uma interferência geralmente tem duração o suficiente para afetar um conjunto de *bits*, sendo que o número de *bits* afetados depende da taxa de transmissão e duração do ruído. Isto nos faz introduzir um dos conceitos mais importantes na detecção e correção de erros: **redundância**. Basicamente enviamos alguns *bits* extras junto com os dados, e isto auxilia para que o receptor detecte e corrija *bits* corrompidos.

Indicação de Vídeo

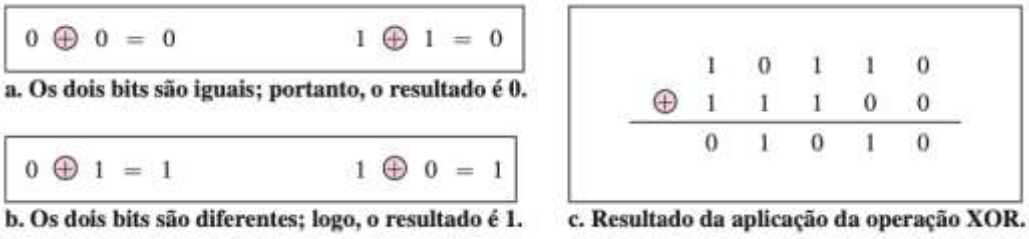
Neste vídeo, vocês aprenderão sobre **Aritmética Modular**, um conceito básico da computação que é essencial para a correção e detecção de erros.

Aritmética - Aula 42 - 7+6=1. Aritmética modular. Disponível em: https://www.youtube.com/watch?v=SswU5Qmc3pg. Acesso em: 19 mai. 2022.

Aritmética - Aula 8 - Congruências (Aritmética Modular). Disponível em: https://www.youtube.com/watch?v=qYqusyUOtF0. Acesso em: 19 mai. 2022.

Como na aritmética de módulo 2, ou binária, nós usamos apenas 0 e 1 para representação, a adição e a subtração produzem os mesmos resultados, logo, usamos a operação XOR (OU Exclusivo), que quando os dois operandos são iguais o resultado é 0, e caso um deles seja diferente, o resultado é 1. Esta operação é ilustrada na Figura 16.

Figura 16: Operação XOR



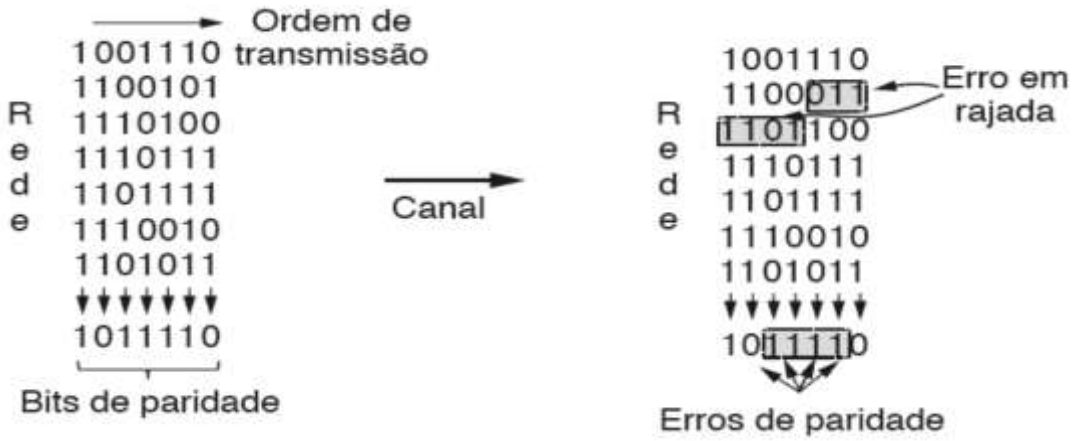
Fonte: FOROUZAN, 2009.

Esta operação é fundamental para muitos algoritmos de detecção e correção de erro que veremos nas próximas seções. A primeira forma de garantir que uma mensagem recebida é a mesma que foi enviada, é através de um *checksum*. Para entendermos o conceito, uma das primeiras formas de garantir algum nível de confiabilidade é o ***bit* de paridade.**

O nível de paridade consiste em acrescentar um único *bit* ao final dependendo do número de *bits* 1 que aparecem na mensagem, sendo par ou ímpar e isto deve ser definido a priori. Por exemplo, se tomarmos 1011010, como temos quatro 1s na palavra, se definirmos a paridade par, colocamos um 0 ao final, resultando em 10110100. Se definirmos pela paridade ímpar, colocamos um 1 no final, resultando em 10110101. Detalhe, se vai ser par ou ímpar, isto tem que ser definido antes.

Com esta simples técnica, conseguimos detectar erros de apenas 1 bit, e se recebermos uma de rajada, como dissemos anteriormente, como podemos tentar descobrir se houve erro?

Figura 17: Bits de paridade em coluna



Fonte: TANENBAUM, 2011.

Podemos calcular os *bits* de paridade em uma ordem diferente daquela de transmissão, ou seja, calcular os *bits* de paridade por colunas (n). Esta técnica de codificação é chamada de entrelaçamento, que consiste de uma técnica geral para converter um código que detecta (ou corrige) erros isolados, em um código que detecta (ou corrige) erros em rajada.

O *bit* de paridade é um modo bem simples, mas ilustra bem a ideia de *checksum*. Contudo, existem outros métodos mais robustos baseados na soma acumulada dos *bits* de dados da mensagem. Geralmente o *checksum* é colocado no final da mensagem, como complemento da função de soma. Assim, no receptor, os erros são detectados somando a palavra inteira, tanto os *bits* da mensagem quanto o *checksum*, e se o resultado for zero, nenhum erro foi detectado. Um exemplo é o *checksum* proposto para a Internet.

Indicação de Leitura

Para quem tiver curiosidade, acesse o link abaixo para ler a proposta de checksum para a Internet.

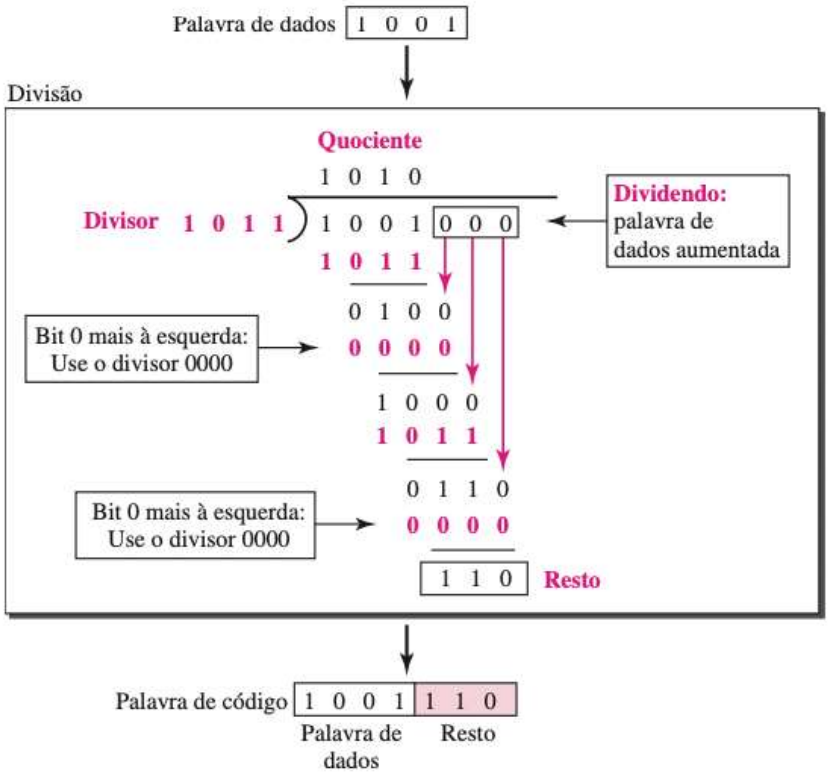
RFC1071 - Computing The Internet Checksum. Disponível em: http://mirror.unpad.ac.id/ietf/rfc/rfc1071.txt.pdf. Acesso em: 19 mai. 2022.

Na prática, um esquema mais forte é utilizado para detecção de erro, é o código de redundância cíclica, ou CRC (do inglês, *Cyclic Redundancy Check*), que é amplamente usado em redes LANs e WANs. Em um código cíclico, se uma palavra for rotacionada ou deslocada para a esquerda, o resultado é outra palavra válida. Tomando 1011000 como exemplo, se deslocarmos para a esquerda, temos 0110001, que também será válida.

O emissor e o receptor concordam num polinômio gerador G(x), em que quanto maior for o seu grau maior será a capacidade de detecção de erros. Para calcular o CRC de um quadro com k bits, que corresponde ao polinômio M(x), o quadro deve ter mais bits do que o polinômio

gerador. Supondo a palavra: 1001, o M(x) gerado é: $1x^3 + 0x^2 + 0x^1 + 1x^0$. Note que o grau desse polinômio é 3 e o tamanho da palavra é 4, ou seja, o grau do polinômio é k - 1. Supondo que o polinômio gerador combinado é: 1011. Assim, realizamos divisões sucessivas como na Figura 17.

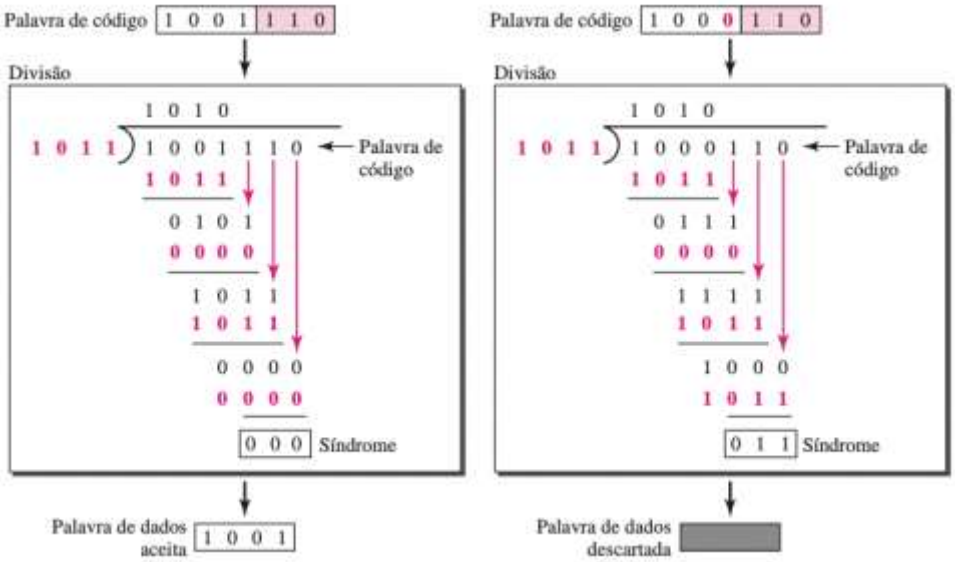
Figura 18: Divisão no codificador CRC (emissor)



Fonte: FOROUZAN, 2009.

Após a divisão, temos como resultado a palavra 1001110, que é o que enviamos para o receptor. Do outro lado da comunicação, pode ser que tenhamos interferência ou não no canal, mas o receptor faz a divisão sucessiva também.

Figura 19: Divisões sucessivas no receptor para caso de sucesso e erro



Fonte: FOROUZAN, 2009.

Como podemos verificar na Figura 19, se o resultado da divisão for 0, significa que a palavra é válida e não contém erros. Já qualquer resultado diferente disso, sabemos que a palavra foi comprometida e será descartada.

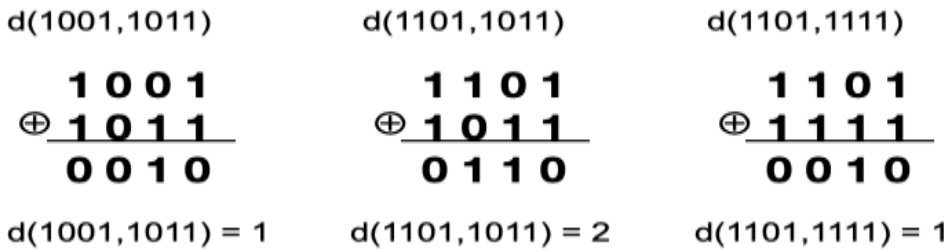
Indicação de Leitura

Para quem tiver curiosidade, o artigo que apresenta o padrão IEEE CRC-32 é discutido no link abaixo.

Koopman, Philip. 32-Bit Cyclic Redundancy Codes for Internet Applications. Disponível em: https://users.ece.cmu.edu/~koopman/networks/dsn02/dsn02_koopman.pdf. Acesso em: 19 mai. 2022.

Outro conceito importante, e talvez um dos mais importantes, é a distância de Hamming. Resumidamente, a distância d(x,y) entre duas palavras é o resultado da operação XOR (⊕) aplicada entre as duas palavras, e a contagem de números de 1s do resultado. Então, podemos inferir que a distância de Hamming é a diferença entre bits correspondentes entre as palavras, podemos verificar como o cálculo é realizado através de exemplos da Figura 20.

Figura 20: Exemplos de cálculos da distância de Hamming para palavras de 4 bits.



Fonte: o autor.

Além deste conceito, a **distância de Hamming mínima** (d_{min}) é uma métrica importante para os algoritmos de detecção e correção de erros. Ela é a menor distância entre todos os pares possíveis, e para determinar este valor, temos que realmente calcular a distância de Hamming entre todas as palavras. Na Figura 21 é apresentado as palavras de código, e as distâncias calculadas entre todas as possibilidades.

Figura 21: Todas as distâncias de Hamming

Palavras de código
000
011
101
110

$d(000, 011) = 2$ $d(000, 101) = 2$ $d(000, 110) = 2$ $d(011, 101) = 2$
$d(011, 110) = 2$ $d(101, 110) = 2$

Fonte: adaptado de FOROUZAN, 2009.

Neste caso, a distância de Hamming mínima é 2. Isto significa que, se ocorrerem dois erros, a palavra de código recebida talvez possa coincidir com uma palavra de código válida e os erros não serão detectados. Mas se ocorrer apenas um erro, é garantido que este erro é detectado. Exemplificando, suponha que enviamos 110, e durante a transmissão tivemos ruído e a mensagem chegou 010, isto será detectado como um erro. Mas se ao invés de 110, a mensagem chegar como 011, foram trocados 2 bits, que é a d_{min}, logo, não é possível detectar o erro, mas a mensagem será lida de maneira errada.

Os **códigos de Hamming** foram desenvolvidos originalmente com $d_{min} = 3$, ou seja, conseguem detectar até dois erros ou corrigir um único erro. Existem codificações capazes de corrigir mais de um erro, mas didaticamente a correção de erro de 1 bit já atende as nossas necessidades. Quando trabalhamos com $d_{min} = 3$, obtemos *n* de $n = 2^m - 1 \Rightarrow 2^3 - 1 \Rightarrow 8 - 1 \Rightarrow 7$ e $k = n - m \Rightarrow 7 - 3 \Rightarrow 4$. Portanto, um código de Hamming C(7, 4). A Figura 22 mostra as palavras de dados e as palavras de códigos.

Figura 22: Código de Hamming C(7,4)

Palavras de dados	*Palavras de código*	*Palavras de dados*	*Palavras de código*
0000	0000000	1000	1000110
0001	0001101	1001	1001011
0010	0010111	1010	101001
0011	0011010	1011	1011100
0100	0100011	1100	1100101
0101	0101110	1101	1101000
0110	0110100	1110	1110010
0111	0111001	1111	1111111

Fonte: FOROUZAN, 2009.

Antes de enviar a mensagem, o emissor preenche com os bits de paridade que é uma combinação da verificação de 3 dos 4 bits da palavra de dados. No receptor, a verificação é realizada usando os bits da palavra e o de paridade, e isto pode indicar um erro em um dos 7 bits recebidos, ou a ausência de erro.

Videoaula 2

Utilize o QR Code para assistir!

Agora, assista ao vídeo em que o professor aborda a ideia de checksum, distância de Hamming e CRC.

Indicação de Vídeo

Neste vídeo, vocês podem ver um pouco mais sobre codificação de Hamming.

How to send a self-correcting message (Hamming codes). Disponível em: https://www.youtube.com/watch?v=X8jsijhllIA. Acesso em: 19 mai. 2022.

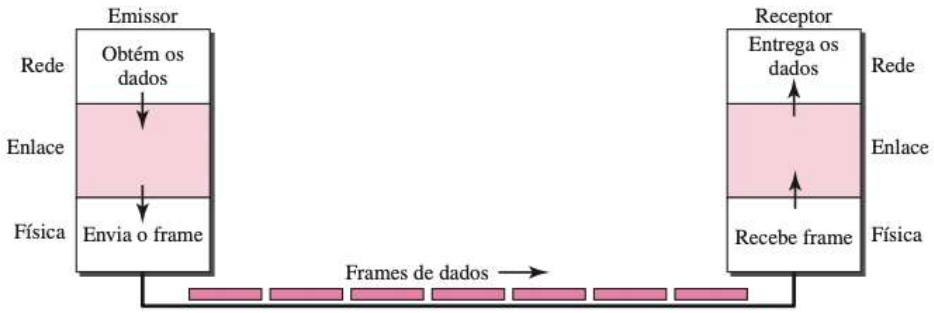
Controle de Fluxos e Erros

O controle de fluxos e erros é a operação mais importante da camada de enlace, em conjunto com as operações que vimos. Nós vimos que a camada de enlace enquadra os *bits*, ou seja, delimita o tamanho das mensagens. Além disso, consegue detectar se houve erro e em alguns casos, até corrigir. Tudo isso, para que agora, controle o fluxo, não permitindo que o emissor sobrecarregue o receptor. Para tanto, alguns mecanismos de controle são implementados. Veremos aqui, a maneira mais simples e teórica, até o protocolo atual. Outra observação é que

trataremos a comunicação como unidirecional, mas nas redes reais, os protocolos são bidirecionais, mas o entendimento de um é facilmente estendido para o outro.

Como já dito, as camadas inferiores provêm serviço para a camada superior, no caso a camada de rede se utiliza da camada de enlace para enviar os dados, e a camada de enlace enquadra e controla o fluxo para depois usar a camada física para enviar os bits. No outro lado, o receptor faz a mesma coisa, recebe os dados e desestrutura e entrega os dados para a camada de rede. Isto é ilustrado na Figura 23.

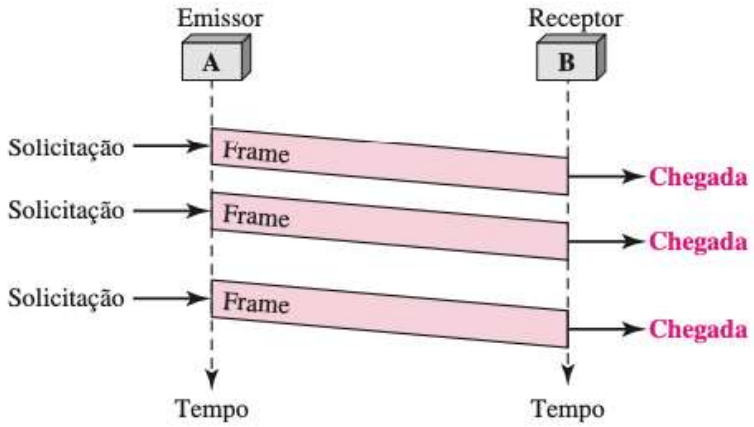
Figura 23: Protocolo mais simples possível



Fonte: FOROUZAN, 2009.

A forma mais simples seria se o canal fosse perfeito, com isto, não haveria necessidade de tratar ou detectar erros, além disso, o receptor poderia receber e tratar todas as mensagens que fossem recebidas, sem encher o *buffer* ou qualquer outra coisa. Isto é retratado na Figura 24, e é um cenário puramente teórico, mas que ajuda a entender o controle de fluxo.

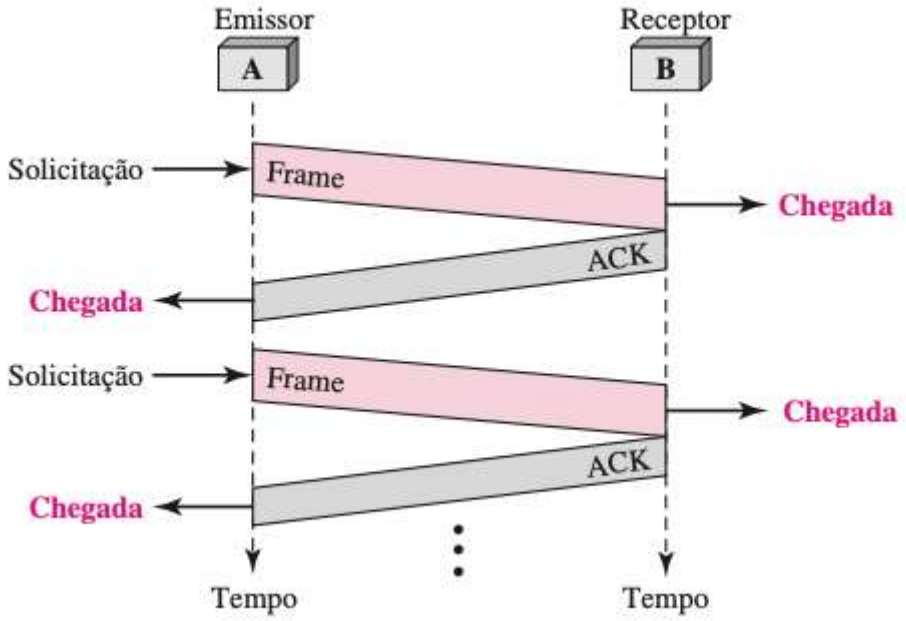
Figura 24: Fluxo de dados



Fonte: FOROUZAN, 2009.

Um dos primeiros pontos que devemos prestar atenção na comunicação é se o receptor está conseguindo tratar as informações adequadamente, ou se não estamos enchendo o *buffer* e desse jeito, fazendo com que o receptor descarte frames. Desta forma, para informar esse excesso, incluímos uma forma de interação entre receptor e emissor, que chamamos de feedback. Este é o protocolo *Stop-and-Wait*, onde o emissor somente envia um frame após receber um ACK (do inglês, *Acknowledgment* - confirmação) do receptor.

Figura 25: Fluxo de dados usando protocolo Stop-and-Wait

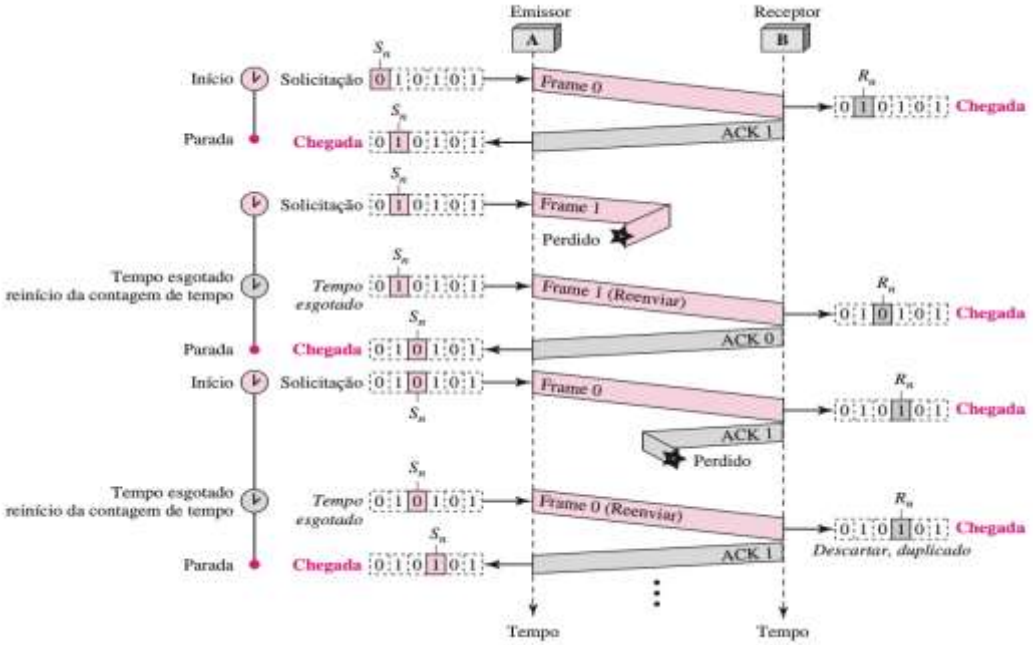


Fonte: FOROUZAN, 2009

Como canais sem ruído inexistem na prática, nós podemos até ignorar o erro, mas temos que ter uma forma de controlar, e o protocolo *Stop-and-Wait* é uma ideia inicial. Um dos problemas que temos com ele é se o receptor receber uma mensagem com erro? E se o emissor não receber uma resposta? Ou se o emissor enviar e o frame não chegar ao receptor? Como tratamos esses cenários? Algumas melhorias são implementadas no *Stop-and-Wait Automatic Repeat Request (Stop-and-Wait ARQ)*.

Quando um receptor não responde ao emissor, significa que o frame estava corrompido. Mas como saber qual frame reenviar? Neste protocolo é adicionado um sequenciamento, assim é possível saber se um frame foi recebido fora de ordem ou duplicado. Além disso, um timer também é adicionado, para que se não houver resposta por parte do receptor, o emissor envia novamente o mesmo frame.

Figura 26: Fluxo de dados usando protocolo Stop-and-Wait ARQ



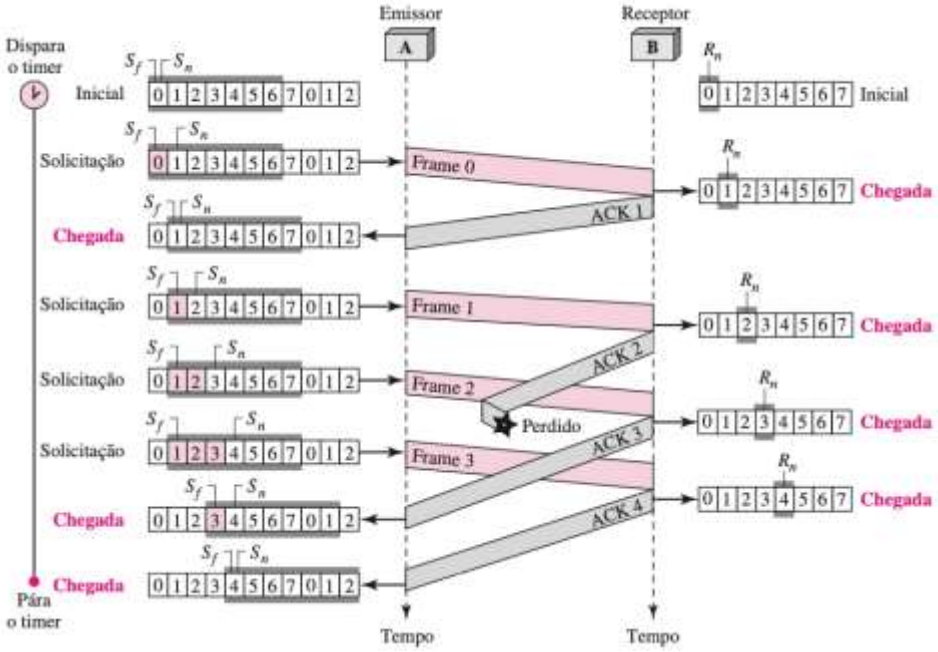
Fonte: FOROUZAN, 2009

O problema desses protocolos até aqui é a ineficiência, pois a sequência de enviar e aguardar uma resposta, mesmo com um timer limitando o tempo, o nosso canal fica ocioso. Para reduzir esse comportamento, adicionamos o mecanismo de *pipelining*, que consiste em iniciarmos o envio do frame, antes mesmo de receber a resposta. Como podemos prever, isso aumenta a complexidade da tarefa, pois temos mais controles necessários. Entretanto, ainda é melhor do que deixar o canal ineficiente.

O *Go-Back-N ARQ* é o primeiro protocolo que funciona com essa premissa, assim, ele permite enviar vários frames antes de receber as confirmações, assim deixamos o canal quase sempre ocupado. É necessário manter uma cópia dos frames enviados, pois se as confirmações não chegarem, são enviados novamente. Mas quantos frames devemos enviar?

Para resolver este problema, neste protocolo é introduzido o conceito de janela deslizante (*sliding window*) onde é definido um intervalo dos números de frames, cabendo ao emissor manter a janela deslizante de transmissão, e o receptor utiliza a janela deslizante de recepção. A janela de transmissão é um conceito abstrato que define um retângulo imaginário de tamanho 2^m - 1 com três variáveis de controle: tamanho da janela, primeiro frame pendente e próximo frame a ser enviado. O nome do conceito é dado, pois a janela desliza uma ou mais posições de acordo com a chegada das confirmações dos frames enviados.

Figura 27: Fluxo de dados usando protocolo Go-Back-N ARQ

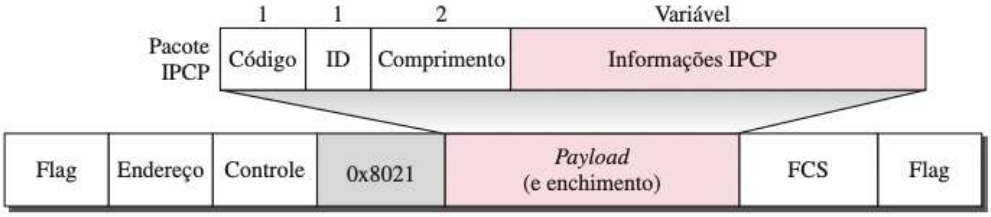


Fonte: FOROUZAN, 2009

A Figura 27 trata-se de um exemplo para o caso no qual o canal de transmissão é confiável, mas o canal de recepção não o é. Nenhum frame de dados é perdido; no entanto, alguns ACKs sofrem atraso e um é perdido. O exemplo também ilustra como as confirmações cumulativas podem ser úteis caso sofram atraso ou sejam perdidas. Após a inicialização, sete eventos ocorrem no lado emissor. Os eventos de solicitação são disparados pela existência de dados na camada de rede; os eventos de chegada são disparados por confirmações da camada física. Observe que, embora o ACK 2 seja perdido, o ACK 3 serve tanto como ACK 2 quanto como ACK 3.

Com isto, finalizamos o estudo de alguns protocolos utilizados na camada de enlace. Dentre os Protocolos Ponto-a-Ponto (PPP), os protocolos de controle de Rede são os de interesse nos nossos estudos. Dentro deles, existe o IPCP (*Internet Protocol Control Protocol)*, que é encarregado de configurar o enlace físico para transportar pacotes IP na Internet. O formato do pacote IPCP encapsulado em um frame PPP é visto na Figura 28.

Figura 28: Pacote IPCP



Indicação de Leitura

Para quem tiver curiosidade, a RFC do protocolo IPCP é apresentada no link abaixo.

TG. McGregor - The PPP Internet Protocol Control Protocol (IPCP). Disponível em: https://datatracker.ietf.org/doc/html/rfc1332. Acesso em: 19 mai. 2022.

Videoaula 3

Utilize o QR Code para assistir!

Agora, assista ao vídeo em que o professor aborda os algoritmos de controle de fluxo e erro, e encerra a segunda unidade.

Encerramento da Unidade

Nesta unidade você aprendeu quais são os objetos das camadas física e camada de enlace. Vimos que a camada física é responsável por transportar os bits através de sinais elétricos ou sinais de luz, ou ainda, através de ondas eletromagnéticas. Cada meio possui sua padronização, contando com interfaces, cabos, antenas e outros dispositivos. Uma vez definida a forma física, temos a camada de enlace que é responsável por fazer o meio de campo entre a camada de rede e a camada física. Ela enquadra os *bits*, detecta e corrige erros e controla todo o fluxo entre o emissor e receptor. A camada física fornece o meio físico de transformar *bits* em sinais, e a camada de enlace utiliza isso para transportar quadros de um lado para o outro.

Com isso, espero ter sido capaz de descomplicar um pouco esse processo de transmitir informações através das redes de computadores e quem sabe, inspirar alguém a pesquisar e aprender um pouco mais sobre. Nosso escopo é maior, e continuaremos na próxima unidade!

Referências

TANENBAUM, Andrew S.; WHETERALL, David J. **Redes de Computadores.** Pearson Prentice Hall, 2011. p. 582.

COMER, Douglas E. **Redes de computadores e internet:** abrange transmissão de dados, ligações inter-redes, web e aplicações. 4. ed. Porto Alegre, RS: Bookman, 2008. 632 p.

KUROSE, James F.; ROSS, Keith W. **Redes de computadores e a internet:** uma abordagem top-down. 3. ed. São Paulo: Addison-Wesley, 2006. 634 p.

FOROUZAN, Behrouz A. **Comunicação de Dados e Redes de Computadores.** 4 ed. AMGH Editora, 2009.


UniFil
EDUCAÇÃO A DISTÂNCIA
UNIFIL.BR