Endereçamento, ARP e Ethernet

Diego Passos

19 de Março de 2018

1 Endereçamento na Camada de Enlace

Como explicado em aulas anteriores, várias tecnologias de camada de enlace proveem um esquema de endereçamento próprio, completamente independente do endereçamento utilizado na camada de rede. Os chamados endereços MAC diferem dos endereços IP, em particular, tanto em formato quanto em utilização.

Enquanto endereços da camada de rede são utilizados no encaminhamento de pacotes ao longo de um caminho completo fim-a-fim, o uso dos endereços MAC se restringe à comunicação local, entre dispositivos diretamente conectados por uma rede em nível 2 — muitas vezes, isso corresponde a nós que estão em uma mesma sub-rede IP, embora isso não seja necessariamente verdade[1].

Em diversas tecnologias, o endereço MAC é composto por 6 octetos (48 bits). Esse formato de endereço é conhecido como EUI-48 ou MAC-48. Ao contrário do que ocorre com os endereços IP, por exemplo, que são determinados pela sub-rede à qual a interface está conectada, os endereços MAC são pré-programados[2] no *hardware* da interface de rede.

A pré-programação do endereço MAC nas interfaces de rede é fundamental para o bom funcionamento das redes de computadores. Um dos objetivos do endereçamento MAC é a portabilidade, ou seja, a capacidade de uma interface ser movida entre redes diferentes sem a necessidade de mudança no endereço. Como os vários usos de endereços pressupõem unicidade, é preciso garantir que cada interface em operação no mundo tenha um endereço próprio, diferente das demais.

Repare como esse objetivo — portabilidade — difere daquele do endereço IP, por exemplo. No IP, endereços não são portáveis porque é necessário respeitar a hierarquia definida pelas sub-redes. Por sua vez, essa hierarquia é importante por conferir melhor escalabilidade ao roteamento da Internet. Na camada de enlace, no entanto, como a comunicação está confinada à rede local — com uma escala, portanto, muito menor que a da Internet como um todo —, escalabilidade não é uma preocupação tão grande, viabilizando a utilização de um **endereçamento plano** .

Mas como garantir essa unicidade dos endereços MAC? A resposta, em primeira análise, é que os fabricantes atribuem endereços únicos a cada interface produzida. É claro que para isso funcionar é necessário que haja alguma coordenação entre fabricantes para que marcas diferentes não produzam interfaces com o mesmo endereço MAC. Nesse ponto entra o IEEE: a entendida controla a atribuição de endereços, determinando faixas de endereços (identificadas por prefixos de três octetos) que podem ser usadas por cada fabricante. Um efeito colateral interessante disso é que é possível identificar o fabricante de um certo equipamento ou interface de rede através prefixo do seu endereço MAC. Vários *softwares* e *sites* na Internet usam e disponibilizam bases de dados com essa associação.

Assim como ocorre com o endereçamento IP, os endereços MAC possuem algumas convenções próprias. Nos endereços EUI-48, por exemplo, o endereço `FF:FF:FF:FF:FF:FF` — ou seja, todos os bits com valor 1 — é reservado ao endereço de *broadcast*. De forma mais geral, endereços cujo bit menos significativo do primeiro octeto seja 1 são considerados endereços *multicast* (*e.g.*, `01-80-C2-00-00-00`), enquanto os demais são considerados *unicast*.

Há, ainda, a distinção entre endereços *universalmente administrados* ou *localmente administrados*. Endereços universalmente administrados são obrigatoriamente únicos e atribuídos de acordo com o processo explicado anteriormente. Já os localmente administrados são atribuídos por administradores de redes locais de forma independente, havendo, portanto, a possibilidade de duplicidade. A separação de um conjunto de endereços localmente administrados permite que o endereço MAC pré-programado de uma interface seja substituído por outro, de acordo com necessidades específicas do administrador

[1] Em aulas posteriores, discutiremos os conceitos de *domínio de colisão* e *domínio de broadcast* que serviram para uma definição mais precisa do que seria essa *comunicação local*. Nessa aula, por simplicidade, continuaremos usando o termo *rede/comunicação local*.

[2] Algumas vezes, diz-se que o endereço MAC é fixo, ou até mesmo *hardcoded* em uma memória ROM. Em dispositivos modernos, no entanto, é muito comum a possibilidade de alteração do MAC via *software* para possibilitar a chamada *clonagem de endereço MAC*. Essa funcionalidade encontra algumas aplicações legítimas que podem ser úteis no dia-a-dia.

do equipamento. Ademais, esses endereços têm sido muito utilizados atualmente para a atribuição de endereços MAC a interfaces de rede de máquinas virtuais. Numericamente, a distinção entre os endereços universalmente e localmente administrados se dá pelo valor do segundo bit menos significativo do primeiro octeto do endereço MAC: o valor 0 indica um endereço universalmente administrado, enquanto o valor 1 denota um endereço localmente administrado.

Em resumo, o formato de um endereço MAC tradicional — EUI-48 — é:

$$bbbbbblm \quad bbbbbbbb \quad bbbbbbbb \quad bbbbbbbb \quad bbbbbbbb \quad bbbbbbbb,$$

onde l determina se o endereço é local e m determina se o endereço é de *multicast*.

2 ARP

A existência de endereços diferentes nas camadas de rede e enlace, embora justificada, resulta em alguns inconvenientes. Considere, por exemplo, a situação em que um roteador recebe um pacote pacote para encaminhamento. Na camada de rede, o endereço de destino do pacote será usado para consultar a tabela de roteamento, que indicará o endereço IP da interface do nó usado como próximo salto no caminho. De posse dessa informação, a camada de rede delega o processo de transmissão do pacote até esse próximo salto para a camada de enlace. Mas qual endereço do nó de próximo salto é passado da camada de rede para a camada de enlace? O endereço usado na camada de rede — *e.g.*, o IP — ou o endereço MAC da interface?

Por conta da desejável independência entre camadas, a camada de rede não deveria conhecer ou manipular os endereços MAC. Da mesma forma, a camada de enlace deveria se preocupar apenas com endereços MAC, ignorando o esquema de endereçamento utilizado na camada de rede. Na prática, no entanto, a necessidade de interfaceamento entre as camadas — ilustrada no exemplo anterior — acaba trazendo a necessidade de algum tipo de **mapeamento entre os endereços** das duas camadas. Esse mapeamento é alcançado através da introdução de um protocolo específico chamado *Address Resolution Protocol*, ou simplesmente ARP.

O procedimento do ARP é muito simples. Suponha que um determinado nó A conhece o endereço IP da interface de outro nó B e, com base nisso, deseja determinar o endereço MAC associado. Isso acontece, por exemplo, quando A deseja transmitir um pacote destinado ao endereço IP de B. Nesse caso, esse pacote original é colocado em espera até que a tradução de endereços seja concluída.

Para isso, A gera um pacote *denominado* ***ARP Query***, contendo o endereço IP a ser resolvido — neste exemplo, o endereço IP de B. O *ARP Query* é encapsulado em um quadro de *broadcast* na camada de enlace, ou seja, com endereço de destino `FF:FF:FF:FF:FF:FF`. A razão para utilização do endereço MAC *broadcast* como destino do quadro é muito simples: a resposta a uma consulta ARP é (geralmente) gerada pelo próprio nó cujo endereço MAC desejamos descobrir. Como **desconhecemos o endereço MAC desse nó**, utilizamos o endereço MAC *broadcast* em substituição.

Por se tratar de um quadro destinado ao endereço de *broadcast*, espera-se que o *ARP Query* seja recebido por todas as interfaces conectadas à rede local. Em particular, espera-se que o pacote alcance a interface de rede do nó B. Note que, também por conta do endereço *broadcast*, todas as interfaces que recebem o quadro irão processá-lo na camada de enlace. O quadro será, então, identificado como um pacote ARP e, por isso, passado para a implementação do protocolo para processamento posterior. O módulo ARP, por sua vez, verificará o endereço da camada de rede a ser resolvido informado no pacote. Caso o endereço corresponda à interface do nó, esse gerará uma resposta: um **pacote do tipo *ARP Reply***.

O *ARP Reply* contém, entre outras informações, o endereço MAC associado ao IP informado no *ARP Query* e o endereço MAC do nó que originou a consulta — no nosso exemplo, das interfaces de B e A, respectivamente. Ao contrário do *ARP Query*, o *ARP reply* é encapsulado em um quadro *unicast* na camada de enlace. Isso porque, por ser gerado em resposta a uma consulta, já se sabe exatamente a quem enviar a resposta.

Ao receber o *ARP Reply*, o nó de origem da consulta extrai o endereço MAC desejado e conclui o processo de tradução. Se esse processo foi disparado, por exemplo, pela necessidade de transmitir um pacote, o endereço MAC recém-descoberto é utilizado e a transmissão do pacote original prossegue normalmente.

O procedimento de resolução adotado pelo ARP introduz um *overhead* na comunicação. Não só pacotes ARP — essencialmente, tráfego de controle — são transmitidos na rede, consumindo recursos, como também os pacotes de dados que necessitam da resolução de endereços tem sua latência aumentada por dependerem da finalização desse processo. Por esse motivo, o ARP utiliza um esquema de *cache* no qual cada mapeamento aprendido é armazenado em uma estrutura de dados denominada

Tabela ARP. Isso reduz o *overhead* da resolução, em particular para mapeamentos frequentemente utilizados.

É importante enfatizar que a Tabela ARP funciona como um *cache*[3]. Assim, cada entrada é associada com um tempo de expiração, ou TTL, demandando que a mesma seja renovada de tempos em tempos. Embora esse tempo varie de sistema para sistema — sendo inclusive, em alguns casos, configurável —, valores típicos são da ordem de alguns minutos. A opção por expirar e esquecer entradas da tabela ARP — seguindo a filosofia *Soft State*, comum na Internet — é justificada por dois aspectos principais:

- **Lidar com o dinamismo do endereço IP**. O endereço IP de uma interface pode mudar e a interface que usa um determinado endereço IP também pode ser alterada. Isso é particularmente comum em redes que utilizem endereçamento via DHCP. Assim, a expiração das entradas na tabela ARP permite que informações defasadas sejam esquecidas e substituídas por versões atualizadas.
- **Manter a tabela relativamente pequena**. Nem todo mapeamento é usado frequentemente. Não fosse a expiração, um mapeamento aprendido por conta de um único pacote transmitido em um passado distante continuaria na tabela ARP indefinidamente. A tendência, nesse caso, seria um crescimento da tabela incentivado por um excesso de entradas de pouca utilidade.

Uma característica marcante do ARP — e de protocolos relacionados à camada de enlace, em geral — é a sua natureza *plug-and-play*. Em outras palavras, o ARP não necessita de configuração prévia. O processo de resolução é disparado automaticamente sob demanda e os mapeamentos aprendidos são gerenciados de forma igualmente automática na tabela ARP. O protocolo, portanto, não requer intervenção manual — embora, em certos casos, administradores de rede possam introduzir entradas manuais na tabela ARP para determinados propósitos específicos.

Outro detalhe relevante sobre o ARP é a sua natureza genérica. Embora sua aplicação mais comum hoje seja na resolução de endereços IPv4 em endereços MAC EUI-48, o ARP pode operar com diversas combinações de endereços da camada de rede e da camada de enlace. Para isso, os pacotes *ARP Query* e *ARP Reply* possuem campos que servem para a identificação dos protocolos e formatos de endereço envolvidos na resolução.

Por fim, note que a descrição do protocolo ARP realizada aqui considera seu cenário de aplicação *tradicional*. Na prática, fabricantes de equipamentos, desenvolvedores de sistemas e administradores de rede acabam empregando o ARP para propósitos diferentes do original.

Um exemplo é o *ARP gratuito*, no qual pacotes do tipo *ARP Reply* ou *ARP Request* são gerados fora de uma situação *normal*. O *gateway* de uma rede, por exemplo, pode gerar um *ARP Reply* gratuito enviado em *broadcast* com o objetivo de ser previamente introduzido na tabela ARP dos demais dispositivos — que provavelmente o utilizarão frequentemente. Clientes DHCP muitas vezes geram *ARP Requests* gratuitos para identificar se o endereço IP ofertado pelo servidor já não está em uso por outro dispositivo. Alguns equipamentos geram *ARP Requests* periódicos para todos os endereços dentro da sua sub-rede IP com o propósito de mapear a rede. Finalmente, algumas configurações exóticas empregam uma técnica chamada de *ARP Proxy* na qual um determinado equipamento responde pelos endereços MAC de várias interfaces.

3 Ethernet

O Ethernet foi criado por Robert Metcalfe na Xerox Parc, no início da década de 1970. Apesar de existir há cerca de 45 anos, o Ethernet é uma tecnologia de grande importância ainda hoje. Trata-se — já há bastante tempo — da tecnologia dominante para redes locais **cabeadas**[4]. Esse domínio é justificado: há várias características que fazem do Ethernet uma solução desejável para esse cenário.

Em primeiro lugar, dispositivos Ethernet são baratos, em comparação com outras tecnologias potencialmente competidoras. Como acontece com diversos produtos populares, há nesse aspecto uma certa influência cíclica: como o Ethernet é popular, há grande demanda por produtos, o que massifica a produção, reduzindo custos e ajudando na popularização dos equipamentos. Mesmo assim, o baixo custo do Ethernet é também — e principalmente — justificado pela simplicidade que este padrão adotou em uma série de aspectos, contribuindo para um *hardware* relativamente simples e, portanto, barato.

[3] Embora vários sistemas permitam a adição manual de entradas permanentes para propósitos de administração.

[4] Essa distinção é importante e ficará mais clara quando estudarmos o Wi-Fi, a tecnologia dominante para redes locais sem fio.

Outra característica que justifica o domínio do Ethernet é histórica: ele foi a primeira tecnologia amplamente utilizada para as redes locais cabeadas, tornando-se um padrão *de facto* — e, posteriormente, um padrão oficial mantido pelo IEEE, chamado de IEEE 802.3. Hoje, uma outra tecnologia concorrente enfrentaria dificuldades de adoção simplesmente porque as redes locais cabeadas que usamos no dia-a-dia são quase exclusivamente Ethernet. Se um usuário final que investe na compra de um computador pessoal, por exemplo, precisar optar entre uma interface de rede Ethernet e outra de uma tecnologia alternativa, quase certamente escolherá a primeira pelo simples fato de ser compatível com redes às quais seu computador será conectado. Da mesma forma, ao projetar uma rede local, um administrador muito provavelmente optará pelo Ethernet por ser a tecnologia que virtualmente 100% da sua base de usuários possui.

Além da já citada simplicidade de projeto, outra característica técnica que justifica a (manutenção da) popularidade do Ethernet foi sua capacidade de evolução ao longo dos anos. Essa evolução incluiu a possibilidade de utilização de diversos meios físicos — que se adéquam a aplicações distintas — e, principalmente, a evolução das taxas de transmissão suportadas: as primeiras versões padronizadas do Ethernet operavam a 10 Mb/s, enquanto o último padrão publicado pelo IEEE prevê até 400 Gb/s.

Essa evolução de taxas permitiu ao Ethernet acompanhar as demandas do mercado e superar a concorrência de outras tecnologias. Em meados da década de 1990, por exemplo, o FDDI (*Fiber Distributed Data Interface*) começou a ganhar espaço ao oferecer a taxa de 100 Mb/s, enquanto o Ethernet alcançava apenas 10 Mb/s. A padronização do chamado *Fast Ethernet*, de 100 Mb/s, tornou o Ethernet competitivo em termos de capacidade, ao mesmo tempo em que seus dispositivos eram mais simples e baratos. Como consequência, o FDDI caiu em desuso, em favor de redes Fast Ethernet.

3.1 Topologia

Desde o seu advento até meados da década de 1990, o Ethernet utilizava exclusivamente uma topologia do tipo barramento, ou seja, um enlace compartilhado — um cabo coaxial, mais especificamente — interconectando todos os nós.

Essa característica de barramento, na qual o sinal transmitido por um nó se propagava para todos os demais nós da rede, foi a motivação para o nome da tecnologia: o prefixo "Ether" se refere ao "Éter luminífero", uma substância hipotética que permearia o Universo na qual ondas eletromagnéticas se propagariam. Embora a teoria do Eter Luminífero tenha sido descartada muitas décadas antes da criação do Ethernet, a suposta substância serviu de inspiração para o nome.

Por se tratar de um meio compartilhado, o barramento Ethernet está sujeito a todos os problemas estudados na aula anterior. Em particular, como o sinal gerado por um transmissor se propaga para todos os demais nós conectados ao barramento, há a possibilidade de colisões. Diz-se, portanto, que todos os nós conectados ao barramento estão em um mesmo **domínio de colisão**.

Na segunda metade da década de 1990, uma topologia alternativa passou a se popularizar. Essa topologia, em estrela, utilizava um dispositivo intermediário especial **chamado de *hub***. Cada dispositivo era conectado a uma das várias portas do *hub* através de um cabo de cobre do tipo par-trançado, utilizando conectores RJ-45. O *hub* era responsável por receber os sinais de cada uma das portas e replicá-los para todas as outras. Era possível, ainda, a interconexão entre dois ou mais *hubs* com o propósito (normalmente) de aumentar o número de portas disponíveis para a conexão de dispositivos à rede.

Repare que, embora externamente a topologia seja outra, internamente o *hub* ainda funciona basicamente como um barramento, já que o sinal transmitido por um dispositivo continua sendo replicado **indiscriminadamente** para todas as demais portas. Com isso, uma topologia Ethernet baseada em *hubs* continua agindo como um único meio físico compartilhado e, portanto, constituindo um único domínio de colisão.

Por conta da possibilidade de colisões, o Ethernet necessitava de um protocolo de acesso múltiplo. Conforme visto na aula anterior, o Ethernet adotou o CSMA/CD como solução. Embora o CSMA/CD seja bastante eficiente, como todo protocolo de acesso múltiplo baseado em acesso aleatório, a probabilidade de colisão aumenta à medida que o número de transmissores ativos cresce. Como consequência, o desempenho da rede cai por conta do aumento de retransmissões e dos tempos de *backoff*.

Na prática, a crescente popularização do Ethernet resultou em redes locais cada vez maiores e que, portanto, tinham desempenho aquém das expectativas. A necessidade de lidar com o problema de desempenho resultante do aumento de colisões em redes maiores acabou levando a uma nova modificação de topologia: embora ainda se tratasse de uma topologia do tipo estrela, o *hub* foi **substituído por um novo equipamento chamado de *switch***.

Externamente, *switches* e *hubs* Ethernet são muito similares. Em suas versões mais comuns, ambos proveem múltiplas portas RJ-45 para a conexão dos demais nós da rede. Entretanto, internamente, o funcionamento dos dois equipamentos é bastante diferente.

Ao contrário de um *hub*, que se limita a replicar o sinal de uma porta para as demais, um *switch* é um **comutador de pacotes**. Um *switch*, portanto, guarda uma série de semelhanças com os roteadores, embora cada equipamento atue em uma camada diferente da pilha de protocolos TCP/IP.

Cada porta de um *switch* é uma interface de rede Ethernet completa e isolada das demais. Portanto, cada cabo conectando um nó a uma porta de um *switch* funciona como um meio de transmissão dedicado — um enlace ponto-a-ponto, em geral *full-duplex* — entre aqueles dois dispositivos. Para que possa haver comunicação entre dois ou mais nós diferentes conectados a portas distintas de um *switch*, este precisa realizar o encaminhamento dos quadros.

Nessa aula, não entraremos em detalhes sobre o funcionamento interno de um *switch* Ethernet — isso será tópico da próxima aula. Por hora, é importante enfatizar que **o uso de *switches* elimina completamente a possibilidade de colisões, desde que, é claro, não haja *hubs* misturados à topologia da rede.**

Este último ponto, alias, justifica uma decisão de projeto interessante do Ethernet. Embora hoje o uso de redes Ethernet baseadas em barramentos ou *hubs* esteja praticamente extinto, as interfaces de rede Ethernet **continuam executando o CSMA/CD, mesmo na comunicação direta entre um nó e a porta do *switch* a qual esteja conectada**. Por que isso ocorre? Justamente porque o padrão garante interoperabilidade com equipamentos antigos, o que inclui a possibilidade de um usuário conectar sua placa de rede Ethernet a um *hub*. Nesse caso, o CSMA/CD é fundamental para viabilizar a comunicação na rede. Repare, no entanto, que quando aplicado a um enlace dedicado o CSMA/CD tem influência quase nula, já que o transmissor irá realizar a detecção de portadora, sempre concluindo que o meio se encontra livre e, com isso, transmitindo imediatamente. O impacto do uso do CSMA/CD no desempenho das redes baseadas em *switch*, portanto, é praticamente zero.

3.2 O Quadro Ethernet

Ao receber um pacote a ser transmitido das camadas superiores, o Ethernet o encapsula em um cabeçalho próprio. Esse cabeçalho é relativamente simples, principalmente se comparado ao de outras tecnologias, como o Wi-Fi.

O padrão especifica tamanhos mínimo e máximo para a carga-útil de seus quadros: 46 e 1500 bytes, respectivamente. Limitar o tamanho mínimo da carga-útil — e, portanto, do quadro como um todo — tem vários propósitos, entre os quais está especificar um atraso mínimo de transmissão. Lembre-se que ao final da aula passada examinamos uma expressão que aproxima a eficiência do CSMA/CD com base nos atrasos de propagação e transmissão. Em particular, essa eficiência é diretamente proporcional ao atraso de transmissão, o que significa que o uso de pacotes muito pequenos — por isso, com baixo atraso de transmissão — resulta em queda da eficiência[5]. Ademais, ao transmitir um quadro muito pequeno, o transmissor passa pouco tempo executando a detecção de colisão. Nesse caso, se houver outra transmissão em curso iniciada por um transmissor muito distante, o mecanismo de **detecção de colisão pode falhar**, fazendo com que a transmissão seja completada com o transmissor acreditando que essa foi bem sucedida. Para evitar esse tipo de caso, o padrão especifica **tanto um limite mínimo para o tamanho de pacote, quanto um limite máximo para a distância** entre dois transmissores.

A especificação de um tamanho máximo, por outro lado, tem como uma de suas vantagem aumentar a justiça na distribuição do uso do meio de transmissão entre os vários transmissores. Se não houvesse um limite, um determinado transmissor poderia monopolizar o uso do meio, transmitindo quadros muito grandes. Além disso, quadros grandes são mais propensos à corrupção de bits. Logo, limitando-se o tamanho de um quadro, limita-se a probabilidade de corrupção.

O primeiro campo de um quadro Ethernet é o **preâmbulo**. Preâmbulos são usados por diversos protocolos com o propósito genérico de auxiliar na recepção do quadro. Em geral, preâmbulos são formados por uma sequência padronizada de bits. Entre outras utilidades, o preâmbulo permite a um receptor identificar o momento em que algum transmissor está iniciando a transmissão de um quadro. No Ethernet, o preâmbulo é formado por uma sequência de 8 bytes: os 7 primeiros tem valor `10101010`, enquanto o último tem valor `10101011`.

Em seguida, encontram-se **dois campos de endereço**: destino e origem. Em particular, o endereço de destino é usado pelas interfaces para filtragem de pacotes. Caso o endereço de destino não seja o endereço MAC da interface, o endereço MAC *broadcast* ou algum endereço *multicast* relevante ao nó, o quadro é imediatamente descartado[6].

[5] Outro benefício de eficiência que justifica limitar o tamanho mínimo de quadros é evitar que os cabeçalhos do Ethernet se tornem, percentualmente, um *overhead* representativo.

[6] Interfaces de rede podem, em geral, operar no chamado *modo promíscuo*. Nesse modo, os filtros da interface são desabilitados, fazendo com que todos os quadros recebidos — independentemente do MAC de destino — sejam entregues ao sistema operacional. Isso é útil case deseje-se realizar captura de pacotes com um *sniffer*.

O próximo campo de cabeçalho é o campo **tipo**. Esse campo é similar ao campo **protocolo** do IPv4. Seu propósito é indicar que tipo de pacote está encapsulado no quadro Ethernet. Por exemplo, pode se tratar de um datagrama IPv4, de um datagrama IPv6 ou de um pacote do protocolo ARP. Constantes numéricas padronizadas são definidas para uma série de protocolos, permitindo que, após o desencapsulamento, o Ethernet saiba para quem — isto é, qual módulo do sistema — enviar a carga-útil.

O campo tipo é seguido da carga-útil que, como dito anteriormente, tem limites de tamanho — inferior e superior — bem definidos pelo protocolo.

Após a carga-útil, o Ethernet acrescenta um pequeno *trailer*. Esse *trailer* contém um único campo: um CRC de 32 bits (mais especificamente, o CRC32). Uma questão interessante é: por que o CRC é adicionado na forma de um *trailer* no Ethernet? Em outras palavras, por que não fazer como o IP ou como o UDP, que informam o *checksum* dos seus respectivos pacotes no próprio cabeçalho.

A razão para isso está na maneira como o CRC é calculado. Ao manter o campo CRC no final do pacote, o Ethernet facilita o cálculo da verificação de integridade. Se o CRC não fosse o último campo do quadro, o receptor seria obrigado a mover os bits de CRC para o final o quadro ao processar a verificação.

3.3 Características

O serviço provido pelo Ethernet é dito **sem conexão**. Isso se deve ao fato de que para que um nó comece a usar uma rede Ethernet, basta que ele seja fisicamente conectado ao resto da rede e comece a transmitir e receber quadros. Não é necessário um processo de *handshake*, como ocorre, por exemplo, no TCP.

Além disso, o Ethernet é fornece um serviço **não confiável**. Isso pode parecer contraditório, já que o Ethernet utiliza o CSMA/CD que detecta colisões e realiza retransmissões nesse caso. No entanto, repare que o CSMA/CD do Ethernet não conta com um *ack*. Assim, se um quadro não sofre colisão — ou se ela não é detectada — o transmissor não é capaz de dizer com certeza se o quadro foi entregue íntegro ao receptor. Interferências, por exemplo, podem introduzir erros ao quadro mesmo em ausência de colisões. O Ethernet, no entanto, assume que, na ausência de colisões detectadas, quadros sempre são entregues corretamente — o que nem sempre é verdade.

3.4 IEEE 802.3

Como dito no início da discussão sobre Ethernet, há bastante tempo essa tecnologia é padronizada: um padrão conhecido como IEEE 802.3. Esse padrão evoluiu ao longo dos anos, adicionando taxas de transmissão mais alta e englobando novos meios de transmissão. Hoje, o padrão inclui suporte a redes baseadas tanto em cabos de cobre, quanto em fibra óptica.

Um detalhe interessante é que as diferentes versões do IEEE 802.3 preveem sempre algum tipo de comprimento máximo dos segmentos interconectando os nós. Há vários motivos para isso, entre os quais pode-se citar:

- A necessidade de limitar o tamanho dos cabos interconectando dispositivos, por conta da atenuação sofrida pelo sinal, enquanto esse se propaga. Depois de certa distância, a perda de energia do sinal se torna tão grande que a compreensão da informação por ele transportada é difícil ou improvável. Esse primeiro problema é muitas vezes contornado com o uso de repetidores, embora isso tenha se tornado menos comum com o advento dos *switches*.

- A necessidade de limitar o atraso de propagação nos segmentos conectando dispositivos, de forma que o protocolo de acesso ao meio funcione de maneira correta. Para distâncias longas demais, o CSMA/CD pode resultar em colisões não detectadas, fazendo com que a comunicação não funcione corretamente.