Protocolos de Acesso Múltiplo

Diego Passos

15 de Março de 2018

1 Enlaces de Acesso Múltiplo

Enlaces de comunicação diferem em vários aspectos, permitindo, portanto, várias formas de classificação diferentes. Uma classificação particularmente importante para esta aula é a entre **enlaces ponto-a-ponto** e **enlaces compartilhados**.

Os enlaces ponto-a-ponto — algumas vezes também chamados de **dedicados** — conectam exatamente dois dispositivos. Alguns exemplos tradicionais incluem um enlace PPP (do inglês *Point-to-Point Protocol*) usado, entre outros fins, para o estabelecimento de uma conexão através de linha discada, e um enlace ponto-a-ponto usado para interconectar um *host* ao seu *swtich* Ethernet.

Em oposição, os enlaces compartilhados — também chamados de **enlaces de difusão, enlaces de *broadcast* ou enlaces de acesso múltiplo** — interconectam potencialmente vários dispositivos utilizando um único meio físico compartilhado. Há inúmeros exemplos de tecnologias de comunicação populares que são baseadas em enlaces compartilhados. Entre elas, podemos citar o padrão Ethernet original (que será discutido em mais detalhes em aulas futuras), as transmissões *upstream* no HFC e o padrão IEEE 802.11 (comercialmente conhecido como Wi-Fi).

Embora comuns, os enlaces compartilhados apresentam desafios adicionais em comparação aos enlaces ponto-a-ponto. Uma maneira intuitiva de entender esses desafios é fazendo a analogia entre um enlace compartilhado e um grande grupo de pessoas conversando em um ambiente fechado. Imagine, por exemplo, um restaurante cheio. O som das vozes das pessoas que conversam se propaga através do ar, meio esse, portanto, compartilhado por todos no ambiente. Suponha que você esteja tentando conversar com um amigo, mas o barulho é tão intenso que impede que seu amigo compreenda o que você está dizendo. Esse "barulho", na verdade, é simplesmente das vozes de outras pessoas que estão, também, tentando falar naquele momento. O fato de haver múltiplos "transmissores" introduzindo seus sinais no meio de transmissão compartilhado dificulta — ou mesmo impede — a compreensão do que foi dito por parte dos "receptores". O ideal, portanto, é que houvesse algum tipo de **coordenação por parte dos transmissores**, para que suas falas fossem intercaladas de forma a não haver essa situação.

2 Protocolos de Acesso Múltiplo

Da analogia feita na seção anterior, podemos definir um conceito de fundamental importância nessa disciplina. Uma **colisão** ocorre quando, em um meio compartilhado, **dois ou mais transmissores transmitem seus sinais simultaneamente**, fazendo com que esses **cheguem misturados ao receptor** que se torna **incapaz de decodificar os conteúdos das transmissões originais**. Embora isso não seja uma verdade absoluta, colisões comumente resultam na perda dos quadros enviados[1].

Note que as colisões consistem em tempo desperdiçado de uso do enlace. Para um enlace de 100 Mb/s, por exemplo, se 5% do tempo é perdido em colisões, a vazão máxima alcançável no enlace passa a ser de 95 Mb/s — desconsiderando quaisquer outras fontes de ineficiência. As colisões são, portanto, um fenômeno nocivo ao desempenho do enlace. Logo, seria interessante que houvesse alguma maneira de eliminá-las, ou pelo menos reduzir sua probabilidade de ocorrência.

É justamente esse o propósito dos **protocolos de acesso múltiplo**. Esses protocolos são basicamente algoritmos — normalmente distribuídos — que definem regras para o compartilhamento do meio de transmissão entre os nós. Em outras palavras, esses protocolos decidem quando cada um dos nós conectados ao meio de transmissão pode acessá-lo. Embora isso não seja sempre verdade, é comum que o protocolo de acesso múltiplo precise trocar algumas informações de controle entre os

[1] Para efeito desta disciplina, assumiremos que uma colisão sempre resulta na perda dos quadros transmitidos simultaneamente.

vários transmissores e, geralmente, essa troca de informações é feita através do próprio enlace compartilhado. Em outras palavras, não há comunicação fora de banda para o controle do protocolo, característica essa que impõe dificuldades adicionais ao projeto desses protocolos.

Ao longo dessa aula, estudaremos vários exemplos de protocolos de acesso múltiplo e, para cada um deles, discutiremos seus pontos positivos e negativos. Para facilitar essa discussão, definiremos aqui o que seria um *protocolo de acesso múltiplo ideal*. Repare que nenhum dos protocolos estudados será perfeito — no sentido de atender a todos os requisitos especificados aqui. Mas alguns dos protocolos práticos se aproximarão mais desse protocolo idealizado que outros.

A primeira característica desse protocolo idealizado é ter **alta eficiência sob baixas cargas**. Em outras palavras, se há apenas um transmissor interessado em utilizar um enlace de capacidade R b/s, esse transmissor deve obter uma vazão média de R b/s.

Por outro lado, gostaríamos de obter **alta eficiência também sob altas cargas**. Isso significa que se M transmissores quiserem constantemente utilizar o enlace compartilhado, cada um deve obter uma vazão média de $\frac{R}{M}$ b/s. Em outras palavras, **a capacidade do meio deve ser dividida de forma justa entre todos os transmissores e não deve haver ineficiências** que reduzam a parcela de capacidade recebida por cada um.

Outra característica desejável é a **total descentralização do protocolo**. Ou seja, nosso protocolo ideal não deve depender de um nó coordenador centralizado ou da existência de uma base de tempo comum a todos os nós. A ausência de um nó central evita a existência de um ponto único de falha na rede, característica sempre desejável em protocolos de comunicação. Além disso, em um sistema distribuído, como uma rede de comunicação, obter sincronismo de relógio entre os nós tem sempre um custo associado.

Por fim, um último requisito desejável — bastante genérico, é verdade — é a **simplicidade**. Seguindo o princípio KISS, uma filosofia amplamente empregada na Internet, daremos preferência a protocolos que sejam simples, em termos de aspectos como a complexidade computacional, o custo de implantação e a facilidade de implementação.

3 As Famílias de Protocolos de Acesso Múltiplo

Embora outras classificações possam ser utilizadas, neste curso seguiremos a taxonomia empregada pelos autores do livro-texto para organizar nosso estudo dos protocolos de acesso múltiplo. Nessa taxonomia, os protocolos são divididos em três grandes famílias:

- **Particionamento de canal**. Nessa família, a capacidade do canal é dividida em **"pedaços" independentes que podem ser atribuídos estaticamente** a transmissores distintos. O termo genérico "pedaços" é usado aqui propositalmente, já que ele pode assumir significados bastante diferentes de protocolo para protocolo. Exemplos clássicos — estudados em mais detalhes a seguir — incluem a divisão em frequência, a divisão no tempo e a divisão em código. De toda forma, a característica determinante dessa família é que, uma vez atribuído a um transmissor, um pedaço passa a ser de uso exclusivo dele. Assumindo que transmissores respeitem essa divisão, **colisões são impossíveis**.

- **Acesso alternado**. Nessa família, o direito de uso do enlace é alternado entre os vários transmissores. Uma vez que um transmissor ganha o direito ao uso do meio, os demais devem permanecer em silêncio. Assim como na família de particionamento de canal, esse método exclui totalmente a possibilidade de colisões, dado que todos os transmissores respeitem os direitos dos demais. **Ao contrário do particionamento de canal, no entanto, esse método permite que um transmissor que não possui dados a transmitir naquele momento abra mão do seu direito de utilizar o meio em favor de outro nó**.

- **Acesso aleatório**. Nessa família, ao contrário das demais, o enlace não sofre qualquer divisão e a coordenação entre os transmissores é mais fraca. Basicamente, quando um transmissor possui dados a transmitir, ele pode **realizar o acesso ao meio com base apenas em informações disponíveis localmente**. Dada a coordenação mais fraca entre os nós, os **protocolos dessa família ainda estão susceptíveis à ocorrência de colisões**. Por isso, esses protocolos geralmente possuem **mecanismos para a detecção da ocorrência de colisões e, nesse caso, para a sua posterior correção**. O nome *acesso aleatório* advém da inclusão de componentes aleatórias na tomada de decisão de um nó sobre acessar ou não o meio em um dado momento — principalmente após a ocorrência de uma colisão.

Nas seções a seguir, estudaremos em mais detalhes essas três famílias, incluindo exemplos de protocolos práticos pertencentes a cada uma delas.

3.1 Particionamento de Canal

Os protocolos de particionamento de canal são, provavelmente, os de mais fácil compreensão dada a simples atribuição estática de partes da capacidade do canal para cada um dos transmissores. Dos três exemplos clássicos — divisão por frequência, tempo ou código — estudaremos dois nessa aula. O método de particionamento de canal por divisão de código será estudado em aulas posteriores, no contexto das redes sem fio.

O TDMA (do inglês *Time Division Multiple Access*) divide o uso do enlace em janelas de tempo — ou *slots*, em inglês. Essas janelas são, normalmente, de duração fixa e idêntica para todos os transmissores. As transmissões ocorrem em rodadas[2] e, em cada rodada, cada transmissor recebe um ou mais *slots* para transmissão. O mais comum, no entanto, é que cada transmissor tenha exatamente um *slot* reservado para seu uso a cada rodada.

A duração do *slot* geralmente é suficiente para a transmissão de exatamente um quadro de dados. Como os slots são alocados estaticamente, é possível que, em uma dada rodada, um transmissor não tenha dados a transmitir, deixando assim seu *slot* ocioso. *Slots* ociosos são, claramente, uma fonte de ineficiência, já que outros nós podem ter dados a transmitir, não podendo utilizar o meio porque aquele *slot* de tempo não lhes foi atribuído.

O quão bem o TDMA aproxima o protocolo ideal descrito anteriormente? Sob altas cargas — *i.e.*, todos os nós desejam transmitir sempre — o TDMA possui alta eficiência: o enlace é utilizado 100% do tempo e, como não há colisões, o tempo de transmissão não é desperdiçado. Além disso, a divisão da capacidade do canal pelos vários transmissores é perfeitamente justa: se cada um dos M transmissores recebe um *slot* (de mesmo tamanho) por rodada, então cada um transmite por uma fração $\frac{1}{M}$ do tempo, efetivamente obtendo uma taxa de transmissão média de $\frac{R}{M}$.

Sob baixas cargas, no entanto, o TDMA é pouco eficiente. Se, por exemplo, apenas um dos M transmissores tem dados a transmitir, cada rodada será composta por 1 *slot* ativo e $M-1$ ociosos, resultando em uma utilização do enlace de apenas $\frac{1}{M}$. Esse único nó transmissor obterá uma capacidade média de $\frac{R}{M}$, embora não enfrente concorrência de outros nós.

Note ainda que o TDMA requer um grau relativamente alto de sincronismo entre os nós. Isso porque cada nó precisa saber com precisão quando seu *slot* começa e termina, de forma a não atrapalhar seus pares.

Um segundo representante dessa família de protocolos é o FDMA (*Frequency Division Multiple Access*). Como o nome sugere, nesse protocolo os "pedaços" do canal atribuídos a cada nó dizem respeito a faixas frequência. O conceito de faixa de frequência está intimamente ligado à camada física, tópico que não será estudado em detalhes nessa disciplina. Mesmo assim, é possível ter uma compreensão abstrata do seu significado considerando o exemplo dos canais de televisão.

Um aparelho de TV — ou receptor de TV digital — recebe o sinal de diversas emissoras através de um único cabo coaxial. Cada emissora transmite seu sinal continuamente. Logo, há múltiplos sinais sendo transmitidos simultaneamente pelo mesmo cabo coaxial — um meio físico compartilhado. Sendo assim, como o receptor é capaz de separar o sinal da emissora desejada? Os sinais não deveriam colidir? A resposta para essa pergunta é que cada emissora transmite em um **canal diferente**, ou seja, uma faixa de frequência específica e separada das utilizadas por outras emissoras. Independentemente de como isso é implementado na camada física, o fato é que esse uso de faixas de frequência separadas para transmissões diferentes permite a extração do sinal desejado, sem interferência dos demais (idealmente).

Esse mesmo princípio é a base do FDMA. Cada transmissor tem alocado para si um canal diferente, ou seja, uma faixa de frequência que deve usar para transmitir seus dados. Assim como no TDMA — e como todo protocolo de particionamento de canal — essa alocação é feita estaticamente. Transmissores podem usar o canal simultaneamente sem risco de colisão, desde que respeitem a alocação de canais.

Em termos de eficiência, o FDMA possui comportamento similar ao TDMA: **alta eficiência sob altas cargas, mas baixa eficiência sob baixas cargas**. A boa eficiência sob altas cargas é resultado da possibilidade de todos os M transmissores poderem utilizar o meio simultaneamente sem risco de colisão. A baixa eficiência sob baixas cargas decorre da ociosidade de canais alocados, caso certos transmissores não possuam dados a transmitir. Repare que a capacidade de um canal no FDMA corresponde a uma fração da capacidade do meio de transmissão como um todo. Assim, em instantes de tempo em que certos canais ficam ociosos, há desperdício de capacidade.

A similaridade do TDMA e do FDMA em termos de eficiência nos diversos cenários não é uma coincidência. De fato, este perfil de desempenho é uma característica dos protocolos dessa família, ocorrendo também no CDMA, que será estudado em aulas futuras.

[2] É comum que essas rodadas sejam chamadas de **quadros TDMA**, mas nessa disciplina evitaremos essa expressão para não haver confusão com o termo *quadro* usado para denotar um pacote na camada de enlace.

3.2 Acesso Aleatório

Protocolos de acesso aleatório — também chamados de *protocolos baseados em contenção* — são de grande importância no estudo das redes de computadores por serem empregados em diversas tecnologias de grande popularidade, como o Wi-Fi e o Ethernet. Os protocolos dessa família se caracterizam por demandar pouca ou nenhuma coordenação entre transmissores, ao contrário do que ocorre no TDMA e no FDMA, por exemplo. Outra diferença em relação a esses protocolos está no fato de que, quando um transmissor realiza uma transmissão, ele utiliza o canal "inteiro", e não apenas um "pedaço", como nos protocolos de particionamento de canal. Finalmente, uma terceira característica particular dessa família é a possibilidade de colisão: ainda que todos os transmissores respeitem perfeitamente as regras do protocolo, transmissões podem colidir, resultando em desperdício de recursos do enlace. Essa é, de fato, uma das principais fontes de ineficiência dos protocolos de acesso aleatório.

Diante disso, protocolos de acesso aleatório em geral especificam como realizar duas tarefas principais: como determinar que uma colisão ocorreu e como recuperar os quadros perdidos devido a colisões.

3.2.1 Aloha

Um protocolo precursor dessa família é o Aloha. Esse protocolo foi introduzido como solução de acesso ao meio na ALOHANet, uma rede sem fio implantada pela Universidade do Havaí na década de 1970 para interconectar usuários localizados em ilhas diferentes do arquipélago[3]. Na realidade, há duas versões do Aloha: a versão original, conhecida como *Aloha Puro*, e uma otimização, conhecida como *Slotted Aloha*. Embora o Aloha Puro tenha precedido o *Slotted* Aloha, por motivos didáticos discutiremos essa segunda variante primeiro.

Como o nome sugere, no *Slotted* Aloha o tempo é discretizado em *slots*, ou janelas de tempo, de duração fixa. Cada *slot* é suficiente para a transmissão de um único quadro (por simplicidade, vamos assumir aqui que todos os quadros possuem o mesmo tamanho). Nós sempre sabem quando começam e terminam os *slots*, e transmissões sempre são iniciadas no início de *slots*. Em outras palavras: se um nó decide transmitir um quadro, mas encontra-se no meio do *slot* atual, ele necessariamente aguardará até que um novo *slot* se inicie para começar sua transmissão. Por fim, assumiremos também que se dois ou mais nós transmitem em um mesmo *slot*, sempre ocorre colisão e todos os nós envolvidos são capazes de detectá-la (note que ainda não entramos no mérito de como essa detecção é realizada).

Dadas essas características e hipóteses, o funcionamento do *Slotted* Aloha é bastante simples. Quando um nó decide enviar um quadro, ele simplesmente aguarda o início do próximo *slot* e realiza a transmissão, independentemente de quaisquer outros fatores. Essa transmissão pode ter dois resultados:

- **Ausência de colisão**: transmissor considera a transmissão do quadro correspondente encerrada. Se houver outro quadro a ser transmitido, ele pode iniciar a nova transmissão no início do próximo *slot*.
- **Colisão**: nesse caso, o nó detectará que a colisão ocorreu e tentará **retransmitir** o quadro nos *slots* subsequentes como forma de recuperação.

Ao contrário da tentativa original de transmissão de um quadro, que sempre ocorre no início do próximo *slot* de tempo, em uma retransmissão, um nó pode demorar vários *slots* até realizar a próxima tentativa. A cada novo *slot*, o nó decide aleatoriamente se realizará a retransmissão naquele *slot* ou não. Mais especificamente, o *Slotted* Aloha utiliza um parâmetro p que determina a probabilidade de um nó decidir realizar a retransmissão no próximo *slot*. Note, portanto, que nas tentativas de retransmissão o nó pode **adiar o acesso ao meio** — um mecanismo conhecido como ***backoff***.

Mas por que não fazer simplesmente com que os nós tentem a retransmissão no próximo *slot* de tempo em caso de colisão? Por que é necessário — ou, ao menos, vantajoso — introduzir esse *backoff* aleatório? Para responder essas perguntas, consideremos um protocolo alternativo em que, em caso de colisão, o nó sempre realiza a tentativa de retransmissão no início do *slot* subsequente. Repare que, por definição, uma colisão envolve dois ou mais transmissores. Logo, todos os — múltiplos — transmissores que tiveram seus quadros colididos em um dado *slot* realizarão retransmissões no *slot* seguinte. Em outras palavras, tais retransmissões se darão, necessariamente, simultaneamente. Como consequência, as retransmissões também falharão por conta de uma nova colisão.

Repare que quando uma colisão ocorre, ela implicitamente introduz uma **sincronização** nos estados dos nós envolvidos, no sentido de que todos eles terão acabado de ter seus quadros perdidos por

[3] A ALOHANet tem grande importância histórica por ser considerada a primeira rede sem fio (pública) baseada em comutação de pacotes.

colisão, necessitando, portanto, realizar suas respectivas retransmissões. Se todos os nós executassem as mesmas ações determinísticas nesse momento — por exemplo, tentar retransmitir imediatamente ou tentar retransmitir daqui a 10 *slots* de tempo — essa sincronização levaria a uma nova colisão, fazendo com que os transmissores jamais conseguissem se recuperar. É justamente para quebrar esse sincronismo que o *backoff* aleatório serve: o adiamento da retransmissão por um nó dá aos demais nós a oportunidade de transmitirem (sozinhos) seus quadros, enquanto a componente aleatória reduz a probabilidade de que nós diferentes agendem suas retransmissões para um mesmo *slot* de tempo novamente.

Embora simples, o *Slotted* Aloha alcança uma série de requisitos do protocolo de acesso ao meio ideal. Por exemplo, ele é altamente eficiente sob baixas cargas. Em um caso extremo, se apenas um transmissor possui dados a transmitir, ele obtém uma utilização de 100% do enlace, já que sem concorrência não há colisões e, portanto, não há *backoffs*. Além disso, o protocolo é altamente descentralizado em termos das decisões tomadas por cada nó: um transmissor decide ou não acessar o meio com base apenas em informações locais. Por fim, a implantação do algoritmo em si é simples, dependendo apenas da capacidade de geração de números pseudo-aleatórios, algo relativamente comum em sistemas computacionais modernos.

Por outro lado, há também pontos negativos. Em primeiro lugar, há a possibilidade de colisões, especialmente sob altas cargas, o que introduz desperdício de tempo de uso do enlace, reduzindo a eficiência do protocolo. Além disso, em caso de retransmissões, *slots* pode ficar ociosos, mesmo que haja dados a serem transmitidos no enlace (isto é, todos os transmissores podem abrir mão de transmitir em um dado *slot* como parte de seus *backoffs*). Em termos de complexidade de implementação, a discretização do tempo em *slots* comuns a todos os transmissores demanda um certo grau de sincronização de relógio. Por fim, note que ainda que um nó tenha a capacidade de detectar uma colisão **durante a transmissão do quadro**, todo o *slot* será desperdiçado em caso de colisão, porque novas transmissões só podem ser começadas no início do próximo *slot*.

Voltemos um instante à eficiência do *Slotted* Aloha. Dada sua operação simples e considerando algumas simplificações razoáveis, é possível modelá-la matematicamente. Para isso, definiremos a eficiência do protocolo como a fração de *slots* **bem sucedidos a longo prazo**. Aqui, *bem sucedidos* significa que **houve uma transmissão e que não houve colisão**. Por outro lado, *a longo prazo* quer dizer que estamos considerando uma sequência arbitrariamente longa de *slots*. Para essa análise, consideraremos que existem N transmissores *ativos*. Em outras palavras, esses N transmissores sempre possuem quadros a enviar[4], enquanto outros nunca transmitem.

Da descrição do funcionamento do *Slotted* Aloha, pode-se inferir que o protocolo possui basicamente dois estados referentes à transmissão de quadros: ou o nó está no processo de realizar a transmissão original do quadro, ou se encontra tentando realizar uma retransmissão. Note que o comportamento de um nó nesses dois estados é diferente, já que, no primeiro, a transmissão sempre é realizada no próximo *slot*, enquanto, no segundo, a transmissão é realizada no próximo *slot* com probabilidade p. Para simplificar a análise, consideraremos apenas o segundo caso[5]. Em resumo: todos os N nós possuem quadros a transmitir em todo *slot* e o farão com probabilidade p.

O primeiro passo nessa análise é determinar a probabilidade de que um *slot* seja bem sucedido. Isso ocorre quando exatamente um nó transmite e os outros $N-1$ não. A probabilidade de que um dado nó transmita sozinho é dada por:

$$p \cdot \underbrace{(1-p) \cdot (1-p) \cdot \cdots \cdot (1-p)}_{(N-1) \text{ vezes}} = p \cdot (1-p)^{(N-1)}$$

Note, entretanto, que como há N transmissores ativos, qualquer um deles pode ser o transmissor bem sucedido no *slot* em questão. Logo, a probabilidade de que um dado *slot* seja bem sucedido é:

$$P(\text{Slot bem sucedido}) = N \cdot p \cdot (1-p)^{(N-1)}$$

A longo prazo, esse valor corresponde à eficiência do protocolo, conforme definido anteriormente:

$$\text{Efic}(p) = P(\text{Slot bem sucedido}) = N \cdot p \cdot (1-p)^{(N-1)}$$

Repare que a eficiência é, portanto, uma função de N (número de transmissores ativos) e p (probabilidade de retransmissão em um dado *slot*). O valor N é uma característica do enlace, mas p pode, a princípio, ser configurado. Uma pergunta natural, portanto, é: **existe algum valor ótimo** de p?

[4] É comum dizermos que esses transmissores possuem um *backlog infinito*.

[5] Repare que em uma situação de alta carga essa hipótese se torna mais razoável, já que colisões se tornam mais comuns, fazendo com que a maioria das transmissões sejam retransmissões de quadros perdidos.

A resposta para essa questão é "sim". Independentemente do número de transmissores ativos, a função Efic(p) possui exatamente um ponto máximo. Determinar esse ponto é simples, bastando calcular a primeira derivada da função e descobrir onde ela se torna nula:

$$\text{Efic}'(p) = N \cdot (1-p)^{(N-1)} - N \cdot (N-1) \cdot (1-p)^{(N-2)}$$

Logo, queremos determinar p tal que:

$$N \cdot (1-p)^{(N-1)} - N \cdot (N-1) \cdot (1-p)^{(N-2)} = 0$$

Assumindo que $N \neq 0$ e $(1-p) \neq 0$:

$$(1-p) - (N-1) \cdot p = 0$$
$$p = \frac{1}{N}$$

Isso indica que o valor ótimo de p depende do número de transmissores ativos no enlace, o que faz sentido se considerarmos que, com mais transmissores ativos, cada transmissor deve ser mais conservador nas suas tentativas de acesso ao meio.

Com base nesse valor, podemos também calcular a eficiência máxima em função de N. Para isso, basta substituirmos o valor ótimo de p em Efic(p):

$$\text{MaxEfic}(N) = \text{Efic}\left(\frac{1}{N}\right) = N \cdot \frac{1}{N} \cdot \left(1 - \frac{1}{N}\right)^{(N-1)}$$
$$\text{MaxEfic}(N) = \left(1 - \frac{1}{N}\right)^{(N-1)}$$

Uma inspeção mais cuidadosa dessa função mostra que ela é **decrescente**, ou seja, **quanto mais transmissores, menor é a eficiência máxima alcançável**. De fato, podemos calcular o limite dessa função a medida que $N \to \infty$:

$$\lim_{N\to\infty} \text{MaxEfic}(N) = \lim_{N\to\infty} \left(1 - \frac{1}{N}\right)^{(N-1)} = \frac{1}{e} \approx 0,367$$

Em resumo, tudo isso significa que a eficiência do *Slotted* Aloha — que já é baixa para poucos transmissores ativos (*e.g.*, 50% para $N = 2$) — tende a pouco mais de 36% à medida que mais transmissores se tornam ativos. Repare ainda que esses valores de eficiência pressupõem o uso do valor ótimo para a probabilidade p. Conclui-se, portanto, que a eficiência do *Slotted* Aloha sob altas cargas é muito ruim.

Lembre-se, no entanto, que o *Slotted* Aloha é uma otimização do Aloha Puro. Logo, deve-se esperar que o Aloha Puro tenha eficiência ainda mais baixa! O Aloha Puro tinha funcionamento ainda mais simples que a versão *Slotted*. No Aloha Puro, não há qualquer sincronização entre os nós, tampouco o conceito de *slots* de tempo. Quando um nó possui um quadro a transmitir, ele simplesmente efetua a transmissão imediatamente. Em caso de colisão, o nó sorteia um valor de *backoff* aleatório de um determinado intervalo. Ao final do período de *backoff*, o nó realiza a retransmissão, independentemente de quaisquer outros fatores.

Não faremos aqui uma análise matemática do Aloha Puro, como fizemos para a versão *Slotted*. Mas, intuitivamente, note que há **mais possibilidades de colisão** na versão pura: enquanto no *Slotted* Aloha, só podem ocorrer colisões entre quadros que começam a ser transmitidos ao mesmo tempo (*i.e.*, no início do mesmo slot), a falta de sincronização do Aloha Puro faz com que um quadro possa agora colidir com outro cuja transmissão foi iniciada **antes ou depois** da sua. De fato, esse aumento da probabilidade de colisão acaba levando a uma piora da eficiência: uma modelagem similar à feita para o *Slotted* Aloha revela que para altas cargas a eficiência do Aloha Puro tende a apenas 18% à medida que o número de transmissores ativos cresce.

3.2.2 CSMA

Um dos fatores que levam o Aloha Puro a eficiências tão baixas é o fato de que transmissões já em curso podem ser atrapalhadas — *i.e.*, colidir — com transmissões iniciadas posteriormente. Considere o seguinte exemplo: um nó A inicia a transmissão de um quadro que leva, ao todo, 200 ms. Após 198 ms — ou seja, após 99% do quadro ter sido transmitido — outro nó inicia a transmissão de outro quadro. Embora a transmissão de A tenha sido atrapalhada por apenas 1% do tempo, isso já é

suficiente para que todo o quadro seja perdido — lembre-se que na comutação de pacotes, se houver alguma inconsistência, o receptor descarta o pacote inteiro.

Essa situação é justamente o que o CSMA (do inglês, *Carrier Sense Multiple Access*) tenta evitar. O CSMA introduz um procedimento relativamente simples, mas que traz grandes benefícios: o *carrier sense*, ou *detecção de portadora*. Em termos simples, a detecção de portadora consiste em ouvir o meio físico com o objetivo de verificar se há transmissões em curso. O diferencial do CSMA, portanto, pode ser resumido em uma frase: **ouça antes de transmitir**. Se o meio se encontra livre, o nó procede com sua tentativa de transmissão. Caso contrário, o nó executa um *backoff*, adiando sua transmissão.

Repare que a **detecção de portadora não é suficiente para coibir todas as colisões**. Isso ocorre porque qualquer tipo de sinal possui uma velocidade de propagação finita, ao mesmo tempo em que qualquer enlace físico possui uma certa extensão. Combinando esses fatores, conclui-se que existe um atraso não nulo desde o momento em que um nó inicia sua transmissão até o instante em que o sinal efetivamente chega aos outros nós. É possível, portanto, que um nó realize a detecção de portadora e não perceba uma transmissão já em andamento, simplesmente porque o sinal ainda está se propagando desde a origem[6].

Mesmo assim, o CSMA resulta em ganhos representativos de eficiência, se comparado ao Aloha — Puro ou *Slotted* — pela grande redução no número de colisões.

3.2.3 CSMA/CD

O CSMA original possui diversas variantes e otimizações. Nessa disciplina, estudaremos duas: nesta aula veremos o CSMA/CD, enquanto em aulas futuras estudaremos o CSMA/CA.

O CSMA/CD (*Carrier Sense Multiple Access with Collision Detection*) é uma otimização do CSMA utilizada, entre outras tecnologias, no Ethernet. Em relação ao CSMA original, o CSMA/CD traz como diferencial o uso da funcionalidade de *detecção de colisão*: ao mesmo tempo em que transmite, o nó realiza uma verificação que permite detectar rapidamente se houve colisão. Caso uma colisão seja detectada, o nó aborta sua transmissão (quase) imediatamente, evitando desperdiçar mais tempo em uma transmissão que já se sabe ter sido comprometida. Essa otimização aparentemente pequena é responsável por ganhos significativos de eficiência em relação ao CSMA original.

Nesta disciplina, não entraremos em detalhes de como a detecção de colisão é realizada. De fato, a forma exata de funcionamento desse mecanismo depende do meio físico do enlace utilizado. De forma bastante simplificada, o transmissor ouve o meio ao mesmo tempo em que transmite, procurando por um sinal com características fora do esperado.

De toda forma, é importante ressaltar que a detecção de colisão **não é viável em todos os tipos de enlace**. Em particular, sua implementação é difícil em enlaces sem fio, o que **justifica a existência do CSMA/CA**, outra variante do CSMA/CD voltada particularmente para enlaces sem fio.

Como forma de resumir o funcionamento do CSMA/CD e discutir alguns de seus aspectos práticos, utilizaremos como exemplo sua implementação no Ethernet. Uma vez que o quadro a ser transmitido chega à interface, o primeiro passo é a detecção de portadora: o nó transmissor **ouve o meio**, verificando se há outras transmissões em andamento naquele momento. **Se o meio estiver livre**, o nó procede para a **transmissão** em si. Por outro lado, **se o meio se encontra ocupado**, o nó **aguarda que esse se torne ocioso** e, então, começa a transmissão.

Uma vez iniciada a transmissão dos bits pelo meio físico, o transmissor é obrigado a, concomitantemente, realizar a **detecção de colisão**. Em outras palavras, ao mesmo tempo em que transmite seu quadro, o nó verifica se alguma colisão ocorre. Caso não se detecte colisão até o fim da transmissão, o processo é concluído. Por outro lado, quando uma colisão é detectada, a transmissão é abortada e o nó passa a gerar **um sinal de *jamming*** por um tempo padronizado.

Um sinal de *jamming* é uma interferência causada de forma proposital. No caso do Ethernet, a geração do sinal de *jamming* tem como propósito garantir que todos os nós conectados ao enlace tratem as transmissões que colidiram como, de fato, uma colisão[7].

Após o envio do sinal de *jamming*, o transmissor entra em **estado de *backoff***. O nó sorteia um valor k de um intervalo $[0, 2^n - 1]$, onde n é o número de colisões sofridas pelo quadro em questão. O *backoff*, então, consiste em aguardar um tempo equivalente a k vezes o tempo necessário para

[6] Outra possibilidade de falha ocorre quando ambos os transmissores estão perfeitamente sincronizados, realizando a detecção de portadora e posterior transmissão dos seus quadros exatamente ao mesmo tempo.

[7] Lembre-se: a colisão é um fenômeno inerente ao receptor. Isso significa que receptores distintos — a distâncias distintas dos transmissores — podem ter percepções diferentes acerca da ocorrência ou não de uma colisão. Para que isso fique mais concreto, imagine um cenário com três nós, A, B e C, no qual A e B e B e C estão a uma distância x um do outro, enquanto A e C se encontram a uma distância $2x$. Considere que A e B comecem simultaneamente a transmitir e abortem suas transmissões assim que percebam o sinal do outro. Pense em como os respectivos sinais chegam a C e nas possíveis consequências disso.

transmitir 512 bits (o que varia de acordo com a taxa de transmissão utilizada). Findo esse tempo, o nó volta ao passo inicial, começando uma nova tentativa de transmissão.

Note que a janela da qual o nó sorteia k dobra de tamanho a cada nova colisão. Por esse motivo, o *backoff* executado pelo Ethernet é muitas vezes chamado de *backoff exponencial binário*. Embora o valor específico de k seja aleatório, por conta desse processo de aumento do intervalo, os tempos de *backoff* tendem a crescer à medida que novas colisões ocorrem com um mesmo quadro. A motivação para aumentar o intervalo é simples: quanto maior o número de colisões sofridas por um mesmo quadro, maior deve ser o número de transmissores competindo pelo uso do enlace. Assim, para reduzir a probabilidade de novas colisões, é preciso dessincronizar as novas tentativas de acesso ao meio. Isso é feito justamente através do aumento do tamanho do intervalo, que reduz a probabilidade de dois transmissores escolherem um mesmo tempo de *backoff*.

Uma modelagem matemática da eficiência do CSMA/CD vai além do escopo dessa disciplina. No entanto, vários autores já realizaram esse tipo de análise. Em particular, um modelo comumente aceito estima a eficiência do Ethernet sob altas cargas como:

$$\text{eficiência} = \frac{1}{1 + \frac{5t_{\text{prop}}}{t_{\text{trans}}}},$$

onde t_{prop} é o atraso de propagação máximo entre quaisquer dois nós no enlace compartilhado e t_{trans} é o atraso de transmissão de um pacote — por simplicidade, assuma que todos os pacotes são do mesmo tamanho.

Embora essa expressão em si não seja importante nessa disciplina, ela provê uma série de conclusões relevantes sobre o CSMA/CD. Em primeiro lugar, ao substituirmos t_{prop} e t_{trans} por valores típicos usados no Ethernet, por exemplo, obtemos eficiências superiores a 90%[8]. Isso é **várias vezes superior à eficiência obtida com o Aloha Puro**, por exemplo.

Além disso, a **eficiência é inversamente proporcional ao atraso de propagação**. Isso faz sentido, já que quanto maior a distância entre transmissores, maior a probabilidade de haver colisão, mesmo com o emprego da detecção de portadora, e mais tempo será desperdiçado em cada colisão, por conta da detecção de colisão demorar mais também. Esse é um dos motivos que justificam o limite do tamanho máximo de enlaces no Ethernet, como discutiremos em aulas futuras.

Por outro lado, a **eficiência é diretamente proporcional ao atraso de transmissão**. Isso significa que, à medida que as taxas de transmissão das tecnologias que usam o CSMA/CD aumentam, a eficiência do protocolo cai (para um tamanho de quadro fixo). Por outro lado, o aumento do tamanho dos pacotes transmitidos pelo enlace contribui para uma maior eficiência do CSMA/CD.

Finalmente, note que o CSMA/CD não requer sincronismo de relógio entre os transmissores, como ocorria no *Slotted* Aloha, tornando-o uma alternativa não só mais eficiente, como também mais simples e barata de se implementar.

3.3 Acesso Alternado

A última família de protocolos de acesso múltiplo é a dos protocolos de acesso alternado (ou *taking-turns*, em inglês). Uma das motivações para a existência dessa terceira família é dicotomia dos protocolos de particionamento de canal e acesso aleatório em termos de eficiência: enquanto o particionamento de canal é bastante eficiente sob altas cargas, sua eficiência cai sob baixas cargas; o acesso aleatório, ao contrário, tem alta eficiência sob baixas cargas, mas sofre sob altas cargas. Os protocolos baseados em acesso alternado, portanto, objetivam manter alta eficiência tanto em cargas altas, quanto em cargas baixas. Evidentemente, paga-se um preço em troca dessa eficiência, conforme discutiremos à seguir.

3.3.1 *Polling*

Um representante simples dessa família é o *polling*. Esse protocolo requer a figura de um nó especial, muitas vezes chamado de *mestre* ou *controlador*. O mestre possui controle do enlace. Os demais nós, chamados de *subordinados*, precisam obter permissão do mestre para acessar o meio.

A permissão para uso do meio é concedida da seguinte forma. O nó mestre, conhecendo a lista de subordinados, começa indagando-os, um a um, sobre seu interesse em utilizar o enlace naquele momento. Ao receber essa indagação, o nó deve sinalizar se deseja ou não usar o enlace nesse instante. Em caso afirmativo, o subordinado recebe permissão para usar o enlace de forma exclusiva por um determinado intervalo a partir daquele momento. Caso contrário, o controlador indaga o próximo

[8] Exercício sugerido: calcule a eficiência considerando um enlace de 30 metros de distância, operando a 1 Gb/s com pacotes de 1500 bytes.

subordinado. Esse esquema de indagação é cíclico, garantindo a todos os subordinados a oportunidade eventual de utilizar o meio.

Se respeitado pelos subordinados, o esquema de *polling* evita qualquer possibilidade de colisão, garantindo boa eficiência sob altas cargas. Além disso, ao contrário do que ocorre com a alocação estática de recursos utilizada nos protocolos de particionamento de canal, os subordinados têm a opção de abrir mão do seu tempo de transmissão em favor dos demais transmissores caso não possuam dados a transmitir naquele momento. Isso faz com que a utilização do enlace se mantenha alta, mesmo sob baixas cargas.

Em contraponto, o sistema de indagação consome recursos do enlace: a comunicação entre mestre e subordinados é feita através de quadros de controle transmitidos pelo próprio enlace compartilhado, consumindo tempo que deixa de estar disponível à transmissão de dados. Outro problema é a latência: caso um único transmissor esteja ativo no enlace, ele precisará aguardar a indagação — inútil, nesse caso — de todos os outros nós subordinados até que tenha o direito de transmitir seu próximo quadro. Por fim, a dependência do protocolo em relação ao nó mestre cria um ponto único de falha no uso do enlace.

3.3.2 Passagem de *Token*

Outro representante dessa família é o protocolo de passagem de *token*. Ao contrário do *polling*, não há a figura de um nó especial atuando como mestre. O uso do meio é feito através da coordenação entre os próprios transmissores. Para eliminar a possibilidade de colisões, esse protocolo utiliza uma regra simples de controle de acesso ao meio: um nó só pode transmitir enquanto estiver de posse do *token*.

O *token* é um conceito abstrato, cuja implementação prática pode variar de tecnologia para tecnologia. Uma das possíveis implementações do *token* é na forma de um quadro de controle especial. Esse *token* é transmitido de forma cíclica entre os nós da rede.

Uma vez que um nó recebe o *token*, ele verifica se deseja usar o meio naquele momento. A posse do *token* garante ao nó o direito de acesso ao meio durante um período máximo pré-determinado. Após esse período ou se o nó não possui dados a transmitir, ele repassa o *token* para o nó seguinte.

A ausência de um nó mestre é, claramente, uma vantagem do esquema de passagem de *token* em relação ao *polling*. No entanto, os mesmos problemas do *polling* existem também na passagem de *token*, ainda que disfarçados: o *overhead* da indagação é substituído pelo *overhead* de passagem de *token*; a alta latência permanece, porque um nó ainda é obrigado a esperar que a autorização de uso do meio passe por todos os seus pares; e, finalmente, o *token* continua constituindo um ponto único de falha no protocolo[9].

4 Exemplos de Aplicação

Uma das principais mensagens tiradas dessa aula é: não existe um protocolo — ou mesmo uma família de protocolos — perfeito. Cada protocolo possui suas vantagens e desvantagens, fazendo com que a escolha dependa da avaliação das características do cenário de aplicação. Para ilustrar como isso é verdade, encerraremos essa aula listando exemplos de diversas tecnologias que empregam as várias famílias de protocolos estudadas aqui.

Tecnologias de comunicação celular são grandes exemplos práticos do uso de protocolos baseados em particionamento de canal. A divisão por frequência (FDMA), a divisão por tempo (TDMA) e a divisão por código (CDMA, que será estudada em aulas futuras) são utilizadas por certas tecnologias celular. Uma das razões para a aplicabilidade dessa família de protocolos a essas tecnologias é a necessidade de um comportamento determinístico, algo que não seria alcançado, por exemplo, com o uso de protocolos de acesso aleatório.

Os protocolos baseados em acesso aleatório, por outro lado, são muito utilizados em tecnologias de comunicação para redes locais. Como já citado anteriormente, o CSMA — nas suas vertentes CSMA/CD e CSMA/CA, respectivamente — é usado, por exemplo, no Ethernet e no Wi-Fi. Nesse caso, é comum que essas redes não apresentem um número tão elevado de usuários simultâneos[10], fazendo com que a boa eficiência sob baixas cargas seja um fator decisivo.

Já os protocolos de acesso alternado encontram aplicação em vários cenários distintos. O Bluetooth, por exemplo, usado na conectividade sem fio de dispositivos periféricos, usa um protocolo

[9] Repare que o *token* pode ser perdido por vários motivos. Por exemplo, um nó de posse do *token* pode sofrer uma pane antes de ser capaz de passá-lo ao próximo transmissor. Protocolos baseados em passagem de *token*, portanto, necessitam de mecanismos de recuperação de *token* para lidar com esse tipo de evento.

[10] Com a enorme popularização do Wi-Fi, isso pode não ser mais exatamente verdade, como discutiremos em aulas futuras.

baseados em *polling*, onde o dispositivo principal (por exemplo, um *smartphone*) atua como mestre de dispositivos subordinados (como um fone de ouvido). Essa família também encontrou aplicação em um tipo de rede que obteve grande popularidade, chamada de *Token Ring*. Como o nome sugere, trata-se de uma aplicação da passagem de *token* em uma topologia em anel.

Como um último exemplo, considere o DOCSIS (*Data Over Cable Service Interface Specification*). O DOCSIS define um padrão para redes de comunicação de dados utilizando uma infraestrutura híbrida de cabos coaxiais e fibra óptica. É um usado, por exemplo, por empresas do ramo de TV a cabo para aproveitar sua infraestrutura pré-existente para fornecer o serviço adicional de acesso à Internet.

No DOCSIS, um modem instalado na residência dos usuário se comunica com uma central da operadora através de um cabo coaxial. Entretanto, esse cabo coaxial funciona como um grande barramento, interconectando vários usuários em uma mesma vizinhança. Dado que se trata de um grande enlace compartilhado, o acesso ao meio precisa ser coordenado de forma a evitar colisões.

Qual protocolo de acesso ao meio é usado pelo DOCSIS? Uma mistura de dois protocolos baseados em particionamento de canal e de um protocolo de acesso aleatório. A capacidade do enlace é dividida em várias faixas de frequência separadas (o que constitui um FDMA). Certos conjuntos de canais são atribuídos para fins diferentes (*e.g.*, *download* e *upload*) e, em determinados conjuntos, múltiplos transmissores são colocados em um mesmo canal. Logo, cada um desses canais é, na verdade, um "sub-enlace" compartilhado no qual outros métodos de coordenação são necessários.