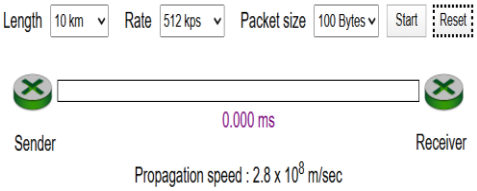
Apresentações: Animações Interativas

Alunos: Eduardo Henrirque e Murillo Fernandes

Transmission versus Propagation Delay

This simple interactive animation illustrates one of the most fundamental concepts in computer networking: transmission delay versus propagation delay. Although this concept is discussed in detail in Chapter 1, an "interactive animation speaks a thousand words". You set the length of the link, the packet size, and the transmission speed; the interactive animation shows the packet being sent from sender to receiver.

Note that for many combinations, the head of the packet reaches the receiver before transmission is finished at the sender.

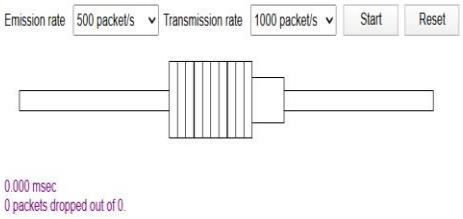


R: Este recurso demonstra os conceitos de atraso de transmissão (tempo para enviar todos os bits de um pacote) e atraso de propagação (tempo que um bit leva para atravessar o enlace). Por exemplo, um pacote de 100 Bytes em um link de 10 km a 512 kbps tem 1,56 ms de transmissão, mas seu cabeçalho chega em apenas 0,036 ms, mostrando que a informação começa a ser recebida antes mesmo do término do envio. Essa diferença é fundamental para entender e otimizar o desempenho de redes de computadores.

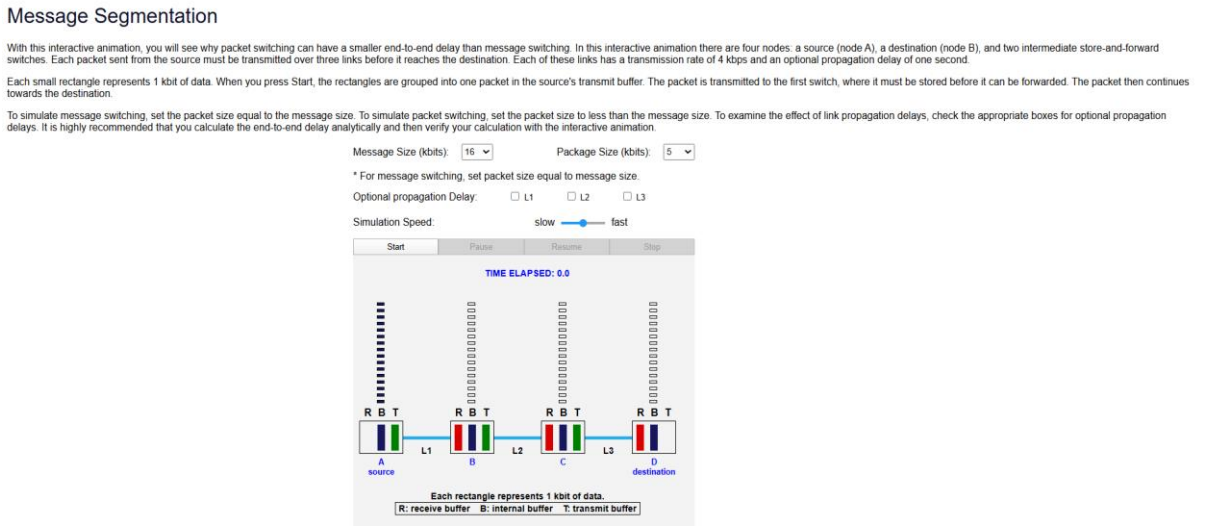
Queuing and Loss Interactive Animation

As we learned in Chapter 1, the most complicated and interesting component of end-to-end delay is queuing delay. In this interactive animation, you specify the packet arrival rate and the link transmission speed. You'll then see packets arrive and queue for service. When the queue becomes full, you'll see the queue overflow--that is, packet loss.

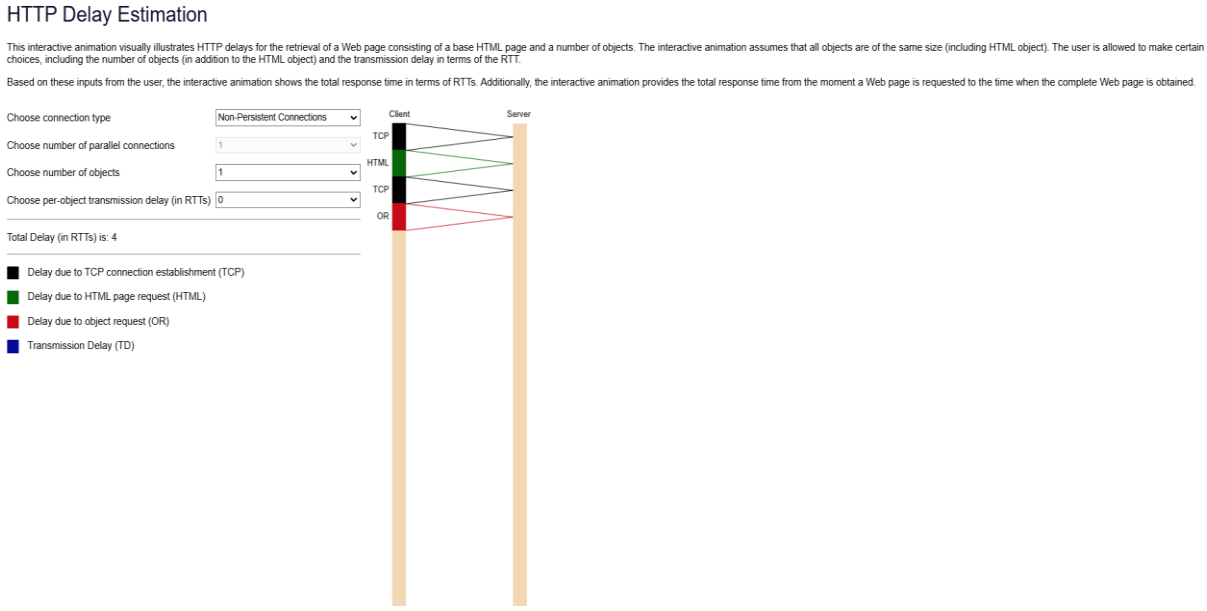
A particularly interesting case is when the emission and transmission rates are the same, for example when both are 500 packets/sec. If you let the interactive animation run for a very long time, you'll eventually see the queue fill up and overflow. Indeed when the two rates are the same (that is, ρ = 1), the queue grows without bound (with random inter-arrival times), as described in the text.



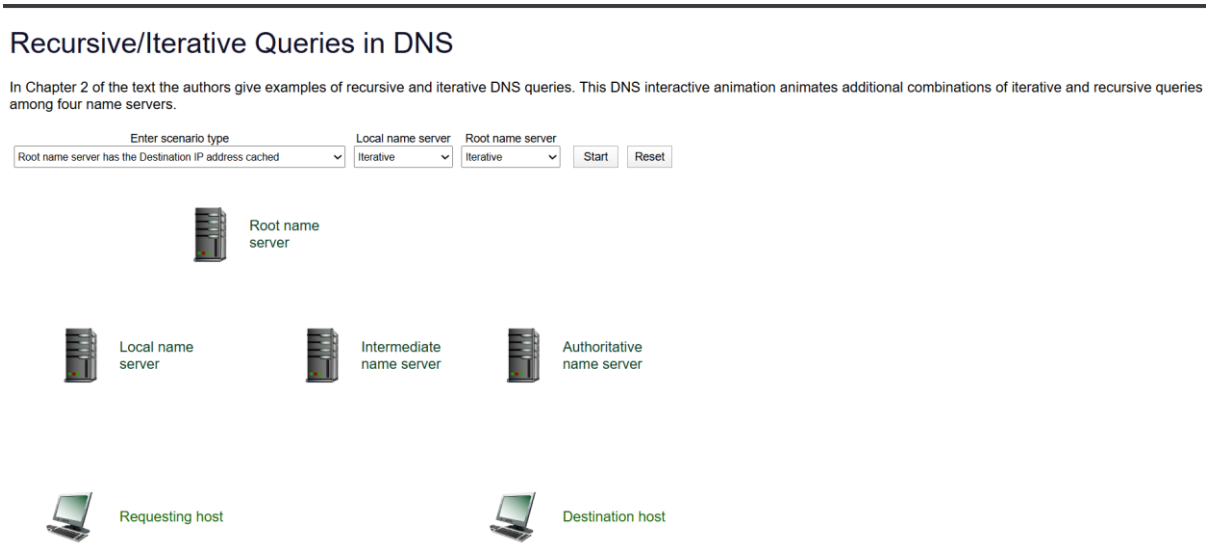
R: Esta animação interativa demonstra como atrasos em fila e perdas de pacotes ocorrem em redes, dependendo da relação entre taxa de chegada e capacidade de transmissão. Quando a chegada de pacotes (ex: 500/seg) é mais lenta que a transmissão (ex: 1000/seg), forma-se uma fila temporária sem perdas; porém, se as taxas se igualam (500/seg em ambos), a fila cresce indefinidamente até causar perda de pacotes por transbordamento. Esse comportamento ilustra como congestionamentos surgem na prática, impactando atrasos e eficiência na comunicação de dados. A ferramenta permite visualizar esses efeitos dinamicamente, reforçando a importância de dimensionar corretamente a capacidade da rede.



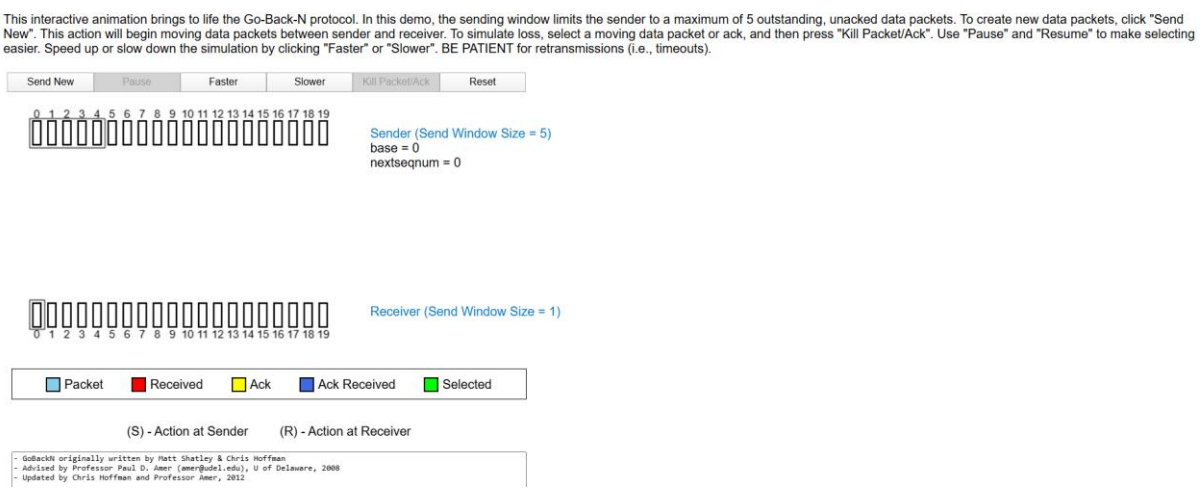
R: Esta simulação demonstra como o cache do navegador reduz o tempo de carregamento ao evitar retransmissões desnecessárias. Quando um cliente faz 70 requisições HTTP 1.1 usando "If-Modified-Since", 40% dos objetos não modificados (28 requisições) retornam status 304 em apenas 10 ms cada (total 280 ms), enquanto os 60% modificados (42 requisições) exigem 21 ms cada (10 ms RTT + 1 ms transmissão + 10 ms RTT, total 882 ms). O tempo total de 1.162 ms mostra a eficiência do cache: sem ele, todas as requisições levariam 21 ms cada, totalizando 1.470 ms - o cache economizou 308 ms (21% de redução). A ferramenta ilustra como cabeçalhos HTTP e armazenamento local aceleram a navegação.



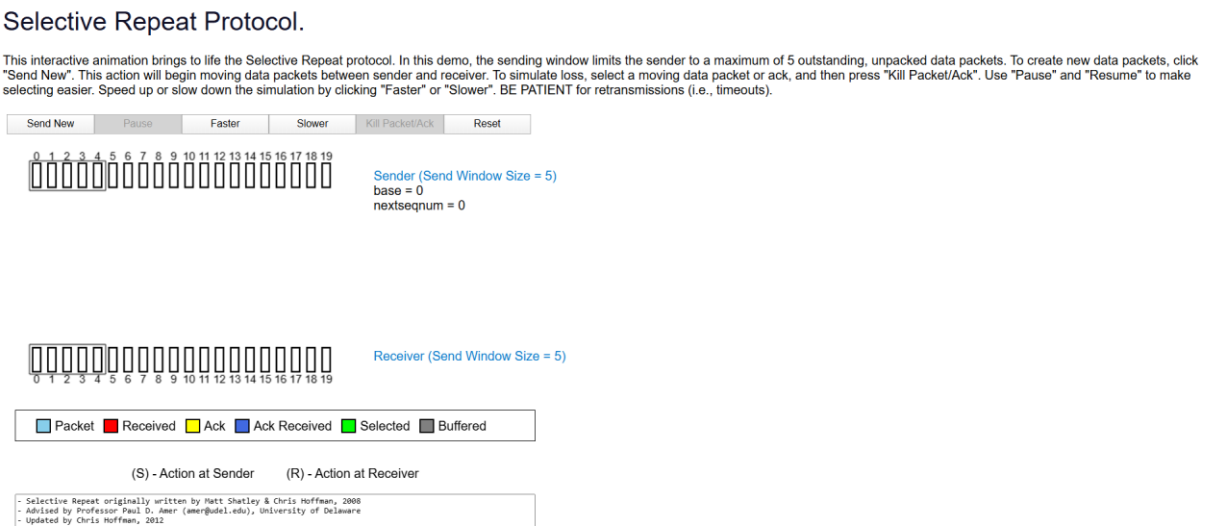
R: Esta ferramenta interativa demonstra como diferentes configurações de conexão HTTP afetam o tempo de carregamento de páginas web. Ao variar parâmetros como tipo de conexão (persistente/não-persistente), número de conexões paralelas e objetos, observa-se que: conexões persistentes eliminam a sobrecarga de estabelecimento repetido de TCP (economizando 1 RTT por objeto), enquanto conexões paralelas permitem carregar múltiplos objetos simultaneamente, reduzindo significativamente o tempo total. Por exemplo, uma página com 4 objetos carrega em 4 RTTs usando conexão persistente, contra 8 RTTs em modo não-persistente.



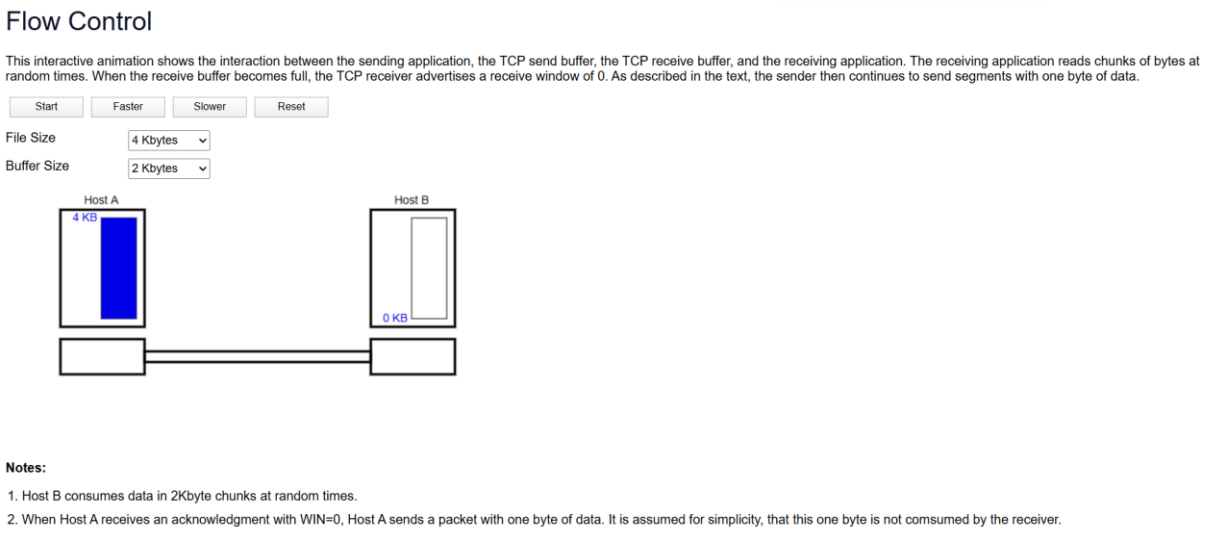
R: Esta animação interativa ilustra a diferença fundamental entre consultas DNS recursivas (onde o servidor local assume toda a cadeia de resolução, retornando apenas a resposta final ao cliente) e iterativas (onde cada servidor referência o próximo na hierarquia, exigindo múltiplas trocas). Ao simular a interação entre servidores local, raiz, intermediário e autoritativo, demonstra-se claramente o trade-off: consultas recursivas simplificam o processo para o cliente às custas de maior latência, enquanto as iterativas distribuem a carga pela rede, reduzindo o tempo total de resposta quando bem configuradas - um equilíbrio crucial para a eficiência do sistema DNS global.



R: Esta animação interativa demonstra o protocolo Go-Back-N (GBN), que controla o envio de pacotes com uma janela deslizante de tamanho fixo (5 pacotes). Quando um pacote é perdido (simulado com "Kill Packet/Ack"), o transmissor retransmite todos os pacotes não confirmados a partir do perdido, mesmo os já recebidos - revelando a principal característica do GBN: simplicidade na implementação (apenas um timer) às custas de retransmissões redundantes. A ferramenta permite visualizar em tempo real como esse mecanismo equilibra eficiência no uso da banda (envio contínuo sem esperar ACKs individuais) com potencial desperdício de recursos, destacando os trade-offs fundamentais no projeto de protocolos de transporte confiáveis.



R: Esta animação interativa ilustra o protocolo Selective Repeat (SR), que melhora a eficiência nas retransmissões usando janelas deslizantes e reconhecimentos seletivos. Diferente do Go-Back-N, o SR retransmite apenas pacotes perdidos ou corrompidos (simulados com "Kill Packet/Ack"), mantendo os demais no buffer do receptor - evitando retransmissões desnecessárias. A ferramenta demonstra como, com uma janela de 5 pacotes, o protocolo otimiza o uso da largura de banda ao permitir transmissão contínua enquanto gerencia perdas de forma seletiva, equilibrando confiabilidade e eficiência na comunicação de dados.

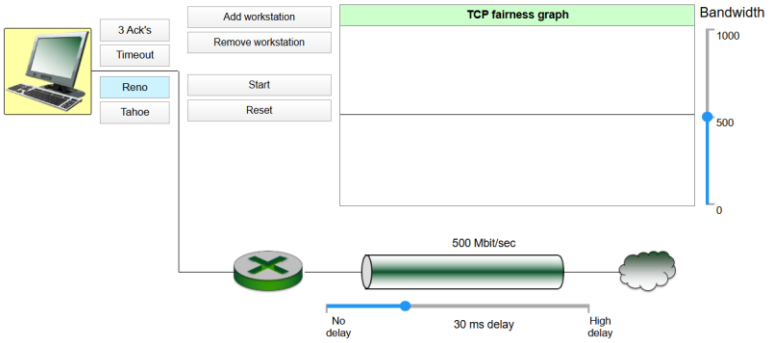


R: Esta animação interativa demonstra o mecanismo de controle de fluxo TCP, onde o tamanho do buffer de recepção (2 KB) regula a transmissão de dados. Quando o buffer do receptor fica cheio, ele anuncia uma janela de recepção zero (WIN=0), forçando o transmissor a enviar pacotes mínimos de 1 byte ("sondas") periodicamente - mesmo sem consumo imediato pelo aplicativo receptor. Esse comportamento previne congestionamentos enquanto mantém a conexão viva, permitindo que a transmissão normal recomece assim que o receptor liberar espaço (ao consumir blocos de 2 KB aleatoriamente). A ferramenta ilustra visualmente como esse equilíbrio dinâmico entre buffers, janelas deslizantes e sondas evita perda de dados e otimiza o uso de recursos na comunicação em rede.

TCP Congestion Control

TCP congestion control is described in Chapter 3 of the text. In this interactive animation, you can view how TCP behaves when multiple clients are sending data over the same link. You may add stations either before or during the simulation. (The default is one station.) When the maximum amount of bandwidth is consumed, all of the sending clients reduce their transmissions rates, depending on the recovery method (Reno or Tahoe).

The green background in the graph shows the total amount of consumed bandwidth. The "3 ack's" button and the "timeout" button trigger the corresponding event for that particular workstation. The "Reno" and "Tahoe" buttons set the recovery method for that particular workstation. The sliders can be used to adjust the speed and the maximum bandwidth.

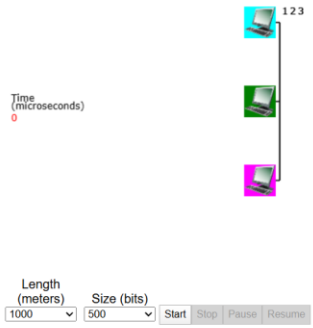


R: Esta animação simula o controle de congestionamento TCP, mostrando como múltiplos fluxos (Reno/Tahoe) ajustam dinamicamente suas taxas ao compartilhar um enlace limitado. Quando a banda máxima é atingida (área verde), os algoritmos reduzem agressivamente as taxas: Tahoe reinicia completamente após qualquer perda (timeout ou 3 ACKs), enquanto Reno mantém parcialmente a capacidade após perdas isoladas (via 3 ACKs). A ferramenta demonstra visualmente o princípio de fairness, onde fluxos estabilizam com taxas equitativas após eventos de congestionamento, e permite manipular parâmetros como banda máxima, delay e eventos de perda para análise.

CSMA/CD

This interactive animation allows you to visualize how transmission time and propagation delay affect CSMA/CD. The interactive animation uses a bus topology (such as with 10Base2) as opposed to a star topology. Similar effects occur with a star topology. The interactive animation assumes a propagation speed of $2*10^8$ meters/sec. The transmission speed is 10Mbps.

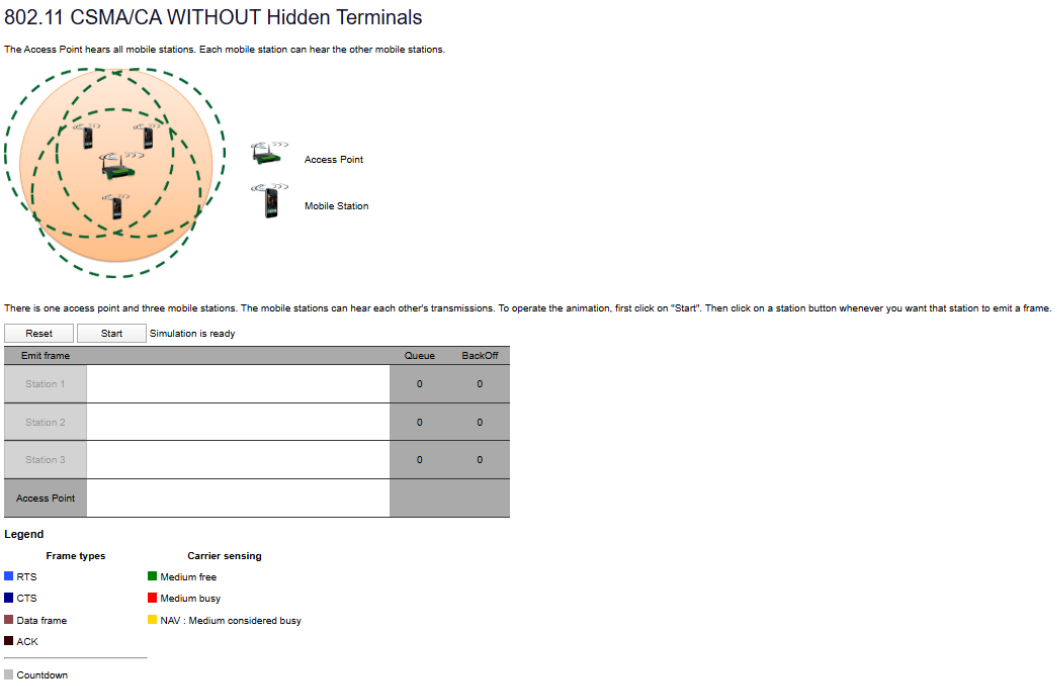
1. Set the parameters: bus length, frame size, and transmission rate.
2. Click on Start.
3. **Click on nodes to generate packets.**



Length (meters) 1000 | Size (bits) 500 | Start | Stop | Pause | Resume

This interactive animation was originally coded by Daniel Brushteyn in 1997 as part of course work at the University of Pennsylvania.

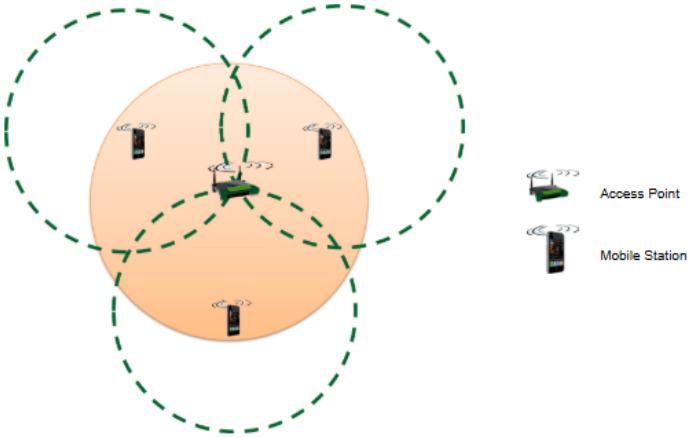
R: Esta animação interativa demonstra o protocolo CSMA/CD usado em redes Ethernet de barramento (como 10Base2), onde o tempo de propagação e o tamanho dos quadros são críticos. Ao ajustar parâmetros como comprimento do barramento (ex: 1000 m) e tamanho do quadro (bits), observa-se como colisões ocorrem quando dois nós transmitem simultaneamente: o protocolo força uma pausa aleatória antes da retransmissão. A ferramenta ilustra visualmente a relação entre velocidade de propagação (2×10^8 m/s) e transmissão (10 Mbps), destacando por que quadros maiores e barramentos mais longos aumentam a janela de vulnerabilidade a colisões - um conceito fundamental para entender Ethernet clássica.



R: Esta animação interativa demonstra o protocolo CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance) em redes 802.11 sem terminais ocultos, onde todas as estações e o ponto de acesso (AP) se ouvem mutuamente. Ao clicar nas estações para gerar quadros, observa-se o mecanismo de detecção de portadora (verificação do meio antes de transmitir), backoff aleatório (contagem regressiva para evitar colisões) e troca de RTS/CTS/ACK para reservar o canal - garantindo transmissões confiáveis mesmo com múltiplos dispositivos competindo pelo mesmo meio sem fio. A ferramenta ilustra visualmente como esses elementos previnem colisões e coordenam o acesso ao meio em redes Wi-Fi tradicionais.

802.11 CSMA/CA WITH Hidden Terminals

The Access Point hears all mobile stations. A mobile station can hear the access point but cannot hear any of the other mobile stations.



There is one access point and three mobile stations. To operate the animation, first click on "Start". Then click on a station button whenever you want that station to emit a frame.

Reset | Start | Simulation is ready

Emit frame		Queue	BackOff
Station 1		0	0
Station 2		0	0
Station 3		0	0
Access Point			

Legend

Frame types	Carrier sensing
RTS	Medium free
CTS	Medium busy
Data frame	NAV : Medium considered busy
ACK	
Countdown	

R: Esta animação demonstra o CSMA/CA com terminais ocultos em redes 802.11, onde estações (ex: 1, 2, 3) não se ouvem mutuamente, apenas o ponto de acesso (AP). O protocolo resolve colisões ocultas usando RTS/CTS: antes de transmitir dados, uma envia *Request to Send* (RTS) ao AP; se o canal estiver livre, o AP responde com *Clear to Send* (CTS), bloqueando virtualmente o meio (via NAV) para outras estações durante a transmissão. O ACK final confirma o sucesso, enquanto falhas (como colisões de RTS) disparam *backoff* aleatório - garantindo acesso justo mesmo quando dispositivos não detectam transmissões paralelas.