

1. Desprezando o tempo de propagação, processamento e enfileiramento, deve-se levar em consideração apenas o tempo de envio dos P pacotes. O tempo de transmissão de *um* pacote se dá por $d = N(L/R)$. No entanto, considerando que cada roteador utilizaria o princípio de *armazenamento e reenvio*, o tempo não será dado apenas por $P * N(L/R)$, uma vez que, durante a transmissão, o pacote seguinte ao anterior já começa a ser recebido a partir do momento em que aquele começa a ser enviado, em cada um dos N enlaces - o conceito de pipelining. Assim, podemos entender que para P pacotes, o tempo de envio se dará pelo tempo de envio fim-a-fim de um pacote somado ao tempo de envio de todos os pacotes remanescentes, enviados em sequência. Logo:

$$D_p = (N + P - 1) * L/R$$

Ao que, por exemplo, considerando, por exemplo $d = 200\text{ms}$, $P = 36000$ e $N = 9$, teríamos:

$$D_p = (9 + 36000 - 1) * 200 = 7201,6 \text{ segundos}$$

2. Apesar de HTTP ser um protocolo sem estado, um cookie pode ser criado durante uma sessão inicial, armazenando dados arbitrários. Em conexões subsequentes, um identificador único (do cookie) pode ser incluído no cabeçalho, identificando uma sessão anterior, e assim recuperando um estado (por exemplo, um login em um site de streaming).

O HTTP/2 introduziu a possibilidade de haver múltiplas requisições e respostas dentro da mesma conexão TCP persistente, de modo que não é necessário esperar todo o conteúdo de uma página ser carregado através de uma única via (sendo outro exemplo de ganho de desempenho através de paralelismo).

DASH é uma abordagem que permite que um mesmo conteúdo seja codificado em diferentes taxas de bits. Para clientes móveis, que frequentemente possuem limitações de sinal (que reduz a confiabilidade ou banda) ou mesmo de consumo de dados, o DASH permite que versões com menor taxa de bits (e, logo, tamanho total) sejam retornadas, permitindo visualização mais contínua (com menos buffering) e com menos consumo.

3. Sim. Ao entrar em um torrent e registrar-se com um rastreador, o usuário começará a receber pedaços do torrent de acordo com as políticas do protocolo, efetivamente começando a receber o conteúdo sem fazer, ao menos inicialmente, nenhum upload. Muito embora o BitTorrent tenha medidas que tornam a transferência igualitária e justa, em um caso remoto onde um usuário entra em um torrent (com seeders ativos) e seja o *único* sem o conteúdo completo, receberia o conteúdo sem ter contribuído com o torrent enviando qualquer dado. No entanto, como dito, esse é um caso remoto e, em geral, usuários contribuem com o torrent enviando dados, ainda que deixem a conexão assim que adquirir o conteúdo completo.

4. No protocolo SR, a janela do receptor pode ficar adiantada em relação à janela da origem, de modo que o receptor envia ACKs para pacotes mesmo que algum pacote anterior não tenha sido recebido, evitando a necessidade de reenvio de pacotes que já foram enviados. A janela da origem é movida para a frente em função dos ACKs recebidos (que em alguns casos podem ser recebidos duas vezes). O uso da janela otimiza o uso da rede, evitando longas esperas por pacotes perdidos, e ao mesmo tempo evita que grandes números de pacotes tenham que ser manuseados (enviados, recebidos e reconhecidos) simultaneamente.

5. Considerando a partida lenta, a conexão iniciaria com o fluxo de 1 MSS, ou seja, 3 KB. Considerando que não há congestionamento da rede, e que o envio é incrementado em 1 MSS a cada ACK, a janela seria estourada em 14 RTTs, ou seja, $14 * 20 = 280\text{ms}$. A esse ponto, o recipiente retornaria um **rwnd** de 0 e a origem passaria a enviar segmentos de 1 byte até o buffer do recipiente esvaziar, para que um novo **rwnd** seja retornado à origem assim que possível, de acordo com o controle de fluxo.

6. Na ocasião de um timeout, o threshold será definido como metade da janela de congestão (cwnd) no momento da detecção da congestão, que por sua vez é determinada como 1 MSS, e o slow start volta a ter efeito. Após as quatro transmissões bem-sucedidas, cwnd será dado por $\text{MSS} * n+1$, onde n é o número de transmissões bem-sucedidas, logo:

$$\text{cwnd} = \text{MSS} * (4+1) \rightarrow 1,5 \text{ KB} * 5 = 7,5 \text{ KB} \quad \text{threshold} = 18 \text{ KB}$$