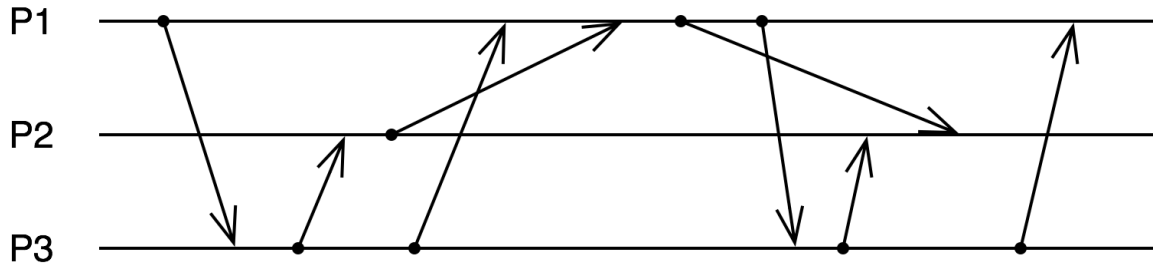
1. O clássico problema dos dois generais especifica que dois exércitos, cada um liderado por um general, se preparam para atacar uma cidade fortificada. Um vale separa os dois exércitos e a única forma dos dois generais se comunicarem é por meio do envio de mensageiros através do vale. Infelizmente, o vale é ocupado pelos defensores da cidade e é possível que um mensageiro seja capturado. Os dois generais concordaram que irão atacar, mas não concordaram sobre a hora do ataque. É necessário que os dois generais ataquem a cidade ao mesmo tempo para terem sucesso, caso contrário um único exército será completamente derrotado. Eles devem, portanto, comunicar para chegarem a um consenso relativamente à hora do ataque, e cada general deve saber que ambos chegaram a um acordo. Mostre que num sistema *assíncrono* tal consenso é impossível de assegurar com qualquer protocolo determinístico com um número fixo de mensagens trocadas entre os generais. Será possível garantir-se o consenso pretendido num sistema distribuído *síncrono*?
2. Um cliente faz uma chamada remota a um método transfere(A, B, €1000) num servidor de transações financeiras, usando semântica at-most-once. No entanto, após enviar o pedido, a ligação falha e não recebe resposta, resultando numa exceção. O cliente pode estar certo de que o método não executou? Pode voltar a tentar (retry) de forma segura?
3. Como podemos solucionar o problema anterior?
4. Por que razão é que todas as operações remotas se traduzem em trocas de byte[]? Como é isto conseguido?
5. No contexto das interações cliente-servidor são muitas vezes usados protocolos do tipo *request-reply* baseados em operações de envio e receção de mensagens. Estas operações de envio e receção de mensagens podem ser descritas em termos semelhantes aos da API de datagramas UDP, apesar de muitas implementações práticas usarem TCP.

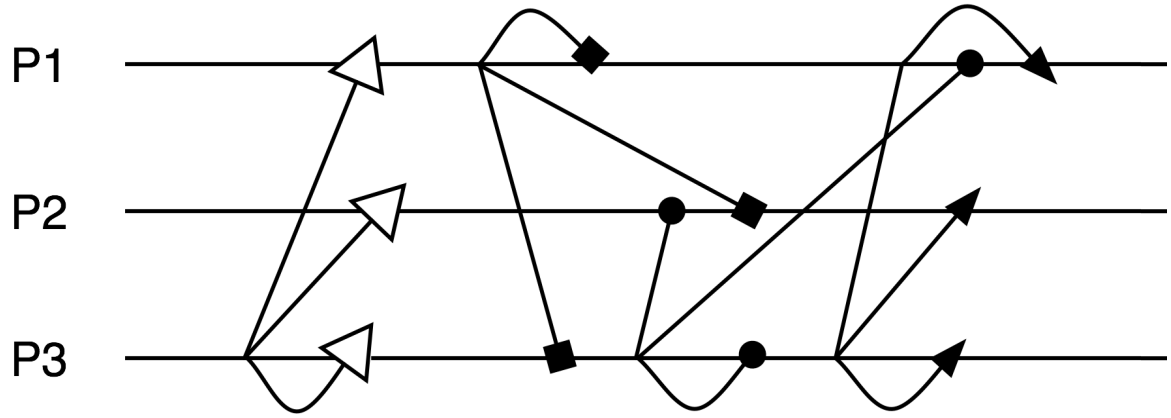
Descreva três vantagens da utilização de UDP para a invocação remota de métodos usando protocolos *request-reply*.

6. E uma razão forte para se usar TCP?
7. Hoje em dia, os *stubs* e os *skeletons* existem conceptualmente, embora sejam gerados dinamicamente durante a execução usando *reflection* em vez de serem pré-compilados. Descreva as funcionalidades que devem suportar, atendendo às questões de comunicação, *marshalling*, invocações e semânticas.
8. O comportamento típico de clientes RPC (ou RMI) consiste em estabelecerem uma ligação ao servidor e executarem longas sequências de chamadas remotas. Partindo desta observação, que melhoramento podemos fazer ao protocolo *request-reply-acknowledge* (RRA) para aumentar o desempenho?
9. Considere a troca de mensagens entre os processos P1, P2 e P3 representada na figura que se segue. Trata-se de eventos de envio e receção de mensagens ponto-a-ponto.

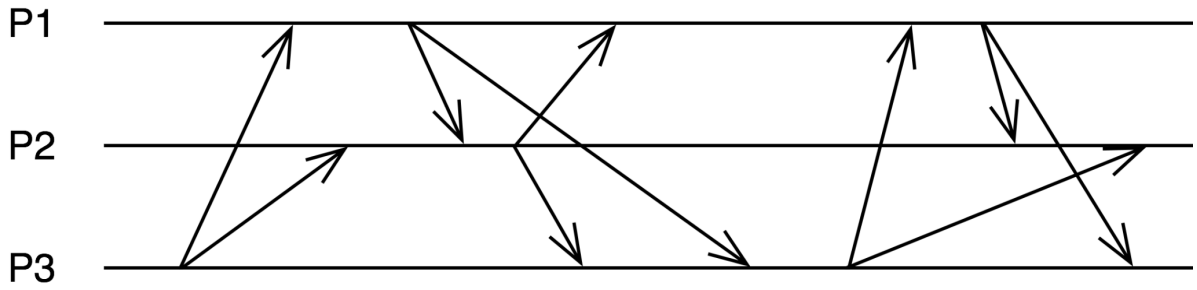


Usando o algoritmo de *vector clocks*, indique na figura os *vector timestamps* em cada ponto de envio e de receção, bem como nas próprias mensagens. Contabilize tanto os eventos de envio como os de receção de mensagens. É possível detetar alguma quebra de causalidade nesta troca de mensagens?

10. O diagrama que se segue ilustra uma troca de mensagens, através de multicast, entre os processos P1, P2 e P3, sendo que as diferentes formas geométricas representam as quatro mensagens distintas e o momento no qual são entregues à aplicação. Que ordenamento de entrega de mensagens é respeitado?



11. Considere a troca de mensagens entre os processos P1, P2 e P3 representada na figura que se segue. Trata-se de eventos de envio e receção de mensagens multicast. Usando o algoritmo de vector clocks, indique na figura os vector timestamps em cada ponto de envio e de receção, bem como nas próprias mensagens. Contabilize apenas os eventos de envio. Indique igualmente receções de mensagens multicast que sejam colocadas na hold-back queue e o ponto em que são entregues à aplicação.



12. Recorde os algoritmos de Cristian e de Berkeley para sincronização de relógios em sistemas distribuídos. Descreva sucintamente as diferenças entre estes dois algoritmos. Explique, em particular, por que razão é enviado o valor absoluto do relógio num caso e o valor do ajuste a efetuar ao relógio no outro caso.
13. Considere que uma máquina A sincroniza o relógio com uma máquina B executando uma iteração do protocolo NTP no modo simétrico. A máquina A enviou o pedido numa mensagem Ma quando o relógio local marcava 14h30m40.500s e a mensagem Mb com a resposta foi recebida pela máquina A à hora local 14h30m40.700s. A mensagem Ma foi

recebida pela máquina B quando o respetivo relógio marcava 14h30m40.520s. A máquina B enviou em Mb o valor de relógio t=14h30m40.640s (juntamente com o instante de receção 14h30m40.520s). Qual será o novo valor de relógio da máquina A?

14. Relativamente à questão anterior, a máquina B pode fazer a leitura local do valor de relógio t em qualquer momento da execução antes de enviar a mensagem Mb à outra máquina? Justifique.
15. Demonstre que a estimativa da diferença entre o valor dos relógios, obtida no modo simétrico do protocolo NTP, se pode obter através da expressão ((T2-T1)-(T4-T3)) / 2.
16. Demonstre que um sistema sincronizado externamente com diferença máxima D está também sincronizado internamente com diferença máxima 2D.
17. Considere as diretivas de *caching* existentes no protocolo HTTP, que permitem controlar quais os objetos que podem ser armazenados em *cache* e configurar esse armazenamento. Descreva sucintamente qual a finalidade de cada uma destas diretivas: no-store, no-cache, private/public, max-age, etag.
18. Construa um fluxograma que sintetize quais as diretivas de *caching* a usar em função de: se uma resposta é reutilizável, se deve ou não ser sempre revalidada, se é armazenável por caches intermédias, se tem um prazo de validade, existência de versões diferentes de cada objeto.
19. Suponha que a Alice pretende enviar um documento D ao Bob, de forma segura. Para tal, a Alice envia ao Bob uma mensagem M = {D}Ks, {{Ks}KBpub, H(D)}KApriv, {A, KApub}KBpub, sendo que a Alice conhece a chave pública do Bob mas o Bob não conhece a chave pública da Alice (e por essa razão a chave pública é enviada em M).
    a. Nestas circunstâncias, a Alice pode estar segura de que exclusivamente o Bob poderá ler o documento? Justifique.
    b. Nesta situação, o Bob pode estar seguro de que o documento D é autêntico e proveniente da Alice? Justifique.
20. Considere um cenário no qual a Alice (A) envia um documento X ao Bob (B). As chaves públicas de ambos são conhecidas por todos previamente. A Alice pretende assegurar a confidencialidade do documento e pretende receber uma confirmação de que o Bob garantidamente abriu e teve acesso a esse documento. Descreva com rigor uma troca de mensagens que permita dar essas garantias.

21. Recorde o protocolo de autenticação de Needham-Schroeder, segundo o qual o processo A pede ao servidor S uma chave secreta K_AB para comunicar com o processo B.
    a. Qual é a finalidade do *nonce* da primeira mensagem?
    b. Qual é a finalidade da terceira mensagem? Justifique sucintamente.
    c. Qual é o objetivo das duas últimas mensagens e como é que esse objetivo é cumprido?
22. Os servidores de DNS (domain name system) podem responder aos pedidos recorrendo a diversos modos de navegação, por forma a traduzir nomes em endereços.
    a. Como podem as pesquisas usando navegação *iterativa* melhorar globalmente o desempenho do DNS? Justifique.
    b. Quais são os benefícios da navegação *recursiva*? Justifique, indicando um exemplo no qual esta navegação seja vantajosa.
23. Considere uma situação na qual se pretende distribuir um ficheiro de 16MB (128Mbit) por um grupo de máquinas usando o protocolo BitTorrent. O ficheiro está dividido em 4 segmentos iguais (também designados por pieces). Uma das máquinas é seeder, outra máquina tem os segmentos {1,2} e as outras duas não possuem qualquer parte do ficheiro. Não há quaisquer outros leechers nem seeders ativos para além destas 4 máquinas. Todas as máquinas têm largura de banda de 8.0Mbit/s de receção mas estão limitadas a 1.0Mbit/s de envio para outras máquinas, sendo que os atrasos de propagação na rede são negligenciáveis.
    a. No caso ideal, qual é o tempo mínimo necessário para que o ficheiro seja distribuído por completo por todas as máquinas neste sistema peer-to-peer? Justifique apresentando os cálculos.
    b. Na prática, relativamente à situação enunciada, a realização de uploads em paralelo para várias máquinas teria alguma influência em conseguir o tempo mínimo calculado na alínea anterior? Justifique.
    c. Na prática, relativamente à situação enunciada, a estratégia de tit-for-tat teria alguma influência em conseguir o tempo mínimo calculado na alínea anterior? Justifique.