Universidade Federal de São Carlos

Trabalho 1 da Disciplina de Sistemas Distribuídos: Sincronização de Clocks

Alunos: Lucas Maziero Russo 321583
Bruno Seiva Martins 321710

Introdução

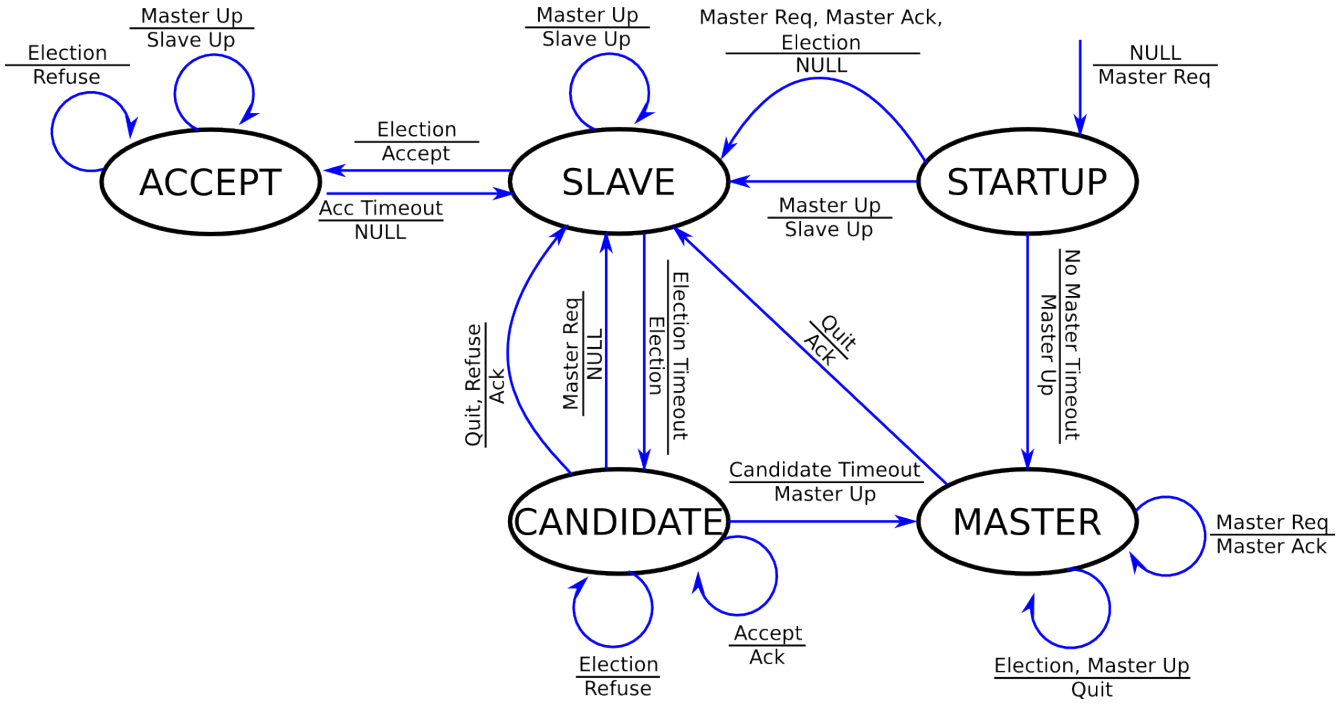
Foi proposta a realização de um sincronizador de clocks distribuídos, escalável (possibilidade de entrada e saída de nós a qualquer momento) e dinâmico (mestre e escravos não pré-determinados).

A abordagem escolhida pela dupla foi baseada na idéia inicial do algoritmo de Berkeley e de sua implementação TEMPO (utilizada na família de sistema operacionais BSD).

Descrição

Basicamente o algoritmo funciona da seguinte forma. É selecionado um mestre dentre os escravos candidatos após a queda (determinado por um timeout) do mestre original. Um caso específico ocorre quando o nó é o primeiro da rede. Nesse caso, o nó emite uma mensagem broadcast na rede local à procura do mestre. Caso ninguém responda (nó é o unico da rede), o nó emite uma candidatura e se torna mestre.

Com o mestre ativo, ocorre um reset do timer dos escravos toda vez que os receberemuma mensagem de sincronização do mestre. Nessa situação, os relógios atuais de todos os escravos são enviados ao mestre para sincronização e posterior envio dos ajustes de volta aos remetentes. A máquina de estados abaixo ilustra o algoritmo e serviu de base para a implementação dos mecanismos de eleição, entrada/saída e sincronização entre os nós da rede.



É importante ressaltar o mecanismo de eleição utilizado. Sempre que um nó deseja participar da rede de sincronização ele envia (via broadcast) uma mensagem de descoberta do mestre. Após certo tempo (timer de descoberta do mestre), se não houver resposta de nenhum nó, o remetente da mensagem emite uma mensagem de candidatura e, novamente, após certo tempo sem resposta o nó original se torna mestre. Caso contrário, o nó recém-chegado se torna escravo e participa passivamente da rede. Uma situação comum ocorre na presença de um timeout de algum escravo (selecionado aleatoriamente entre um mínimo de "intervalo de sincronização" e um máximo pré-estabelecido). No caso, o nó em questão envia uma mensagem de candidatura (via broadcast) aos nós da rede. Todos os escravos destinatários respondem à mensagem retornando ao remetente um accept. Após certo tempo do recebimento de mensagens positivas (accept) ocorre um timeout e o candidato se torna mestre. Vale frisar que situações átipicas (mestre "lento", dois ou mais nós lançam candidatura simultaneamente, entre outras) são tratadas de acordo com a máquina de estados acima de forma a manter a consistência (sempre deve haver apenas um mestre) e robustez (mestre desaparece) do algoritmo.

Por fim, a sincronização dos relógios ocorre a cada um minuto (tempo do mestre). Quando isso ocorre, uma mensagem de requisição de clocks para todos os escrados é enviada. Após isso, os escravos enviam seus relógios atuais e o mestre realiza a sincronização baseada na média dos relógios de todos os escravos mais a do próprio mestre. Além disso, o tempo de propagação da mensagem entre o mestre e os escravos são considerados estimando-se o RTT = (tempo em que o mestre recebeu a resposta - tempo em que o mestre enviou a requisição) / 2. A figura abaixo ilustra o mecanismo descrito: