

Resumo sobre acesso aleatório e revezamento

Nome: Jeferson Gonçalves Noronha Soriano

Matricula: 471110

Acesso aleatório

Quando um nó tem um dado para ser enviado ele vai enviar ele envia na velocidade total do canal sem nenhuma ordem entre os no, o que significa que quando um nó quer enviar um dado ele enviar tudo se necessário pelo canal, agora imagine que 2 no decidem enviar tudo ao mesmo tempo, gerando uma colisão entre os nós, fazendo que eles se destruam, aí que entra os protocolos de acesso aleatório, tratando essas colisões, vamos falar de 2, o ALOHA e o slotted ALOHA

Vamos começar com o **slotted ALOHA** que divide o tempo de transmissão pelo tamanho de quadros, os quadros têm o mesmo tamanho, assim quando um quadro pega um tempo de transmissão, esse tempo será do seu tamanho, aí quando 2 no decidem transmitir nos mesmo quadro de intervalo ocorre a colisão o protocolo resolve da seguinte maneira, os nós tentam se transmitir nos quadros subsequentes até que haja sucesso.

esse protocolo é simples de implementar, um unico no ativo transmite na velocidade integral do canal e descentralizado, mas ocorrem muitas colisões, muitos intervalos ociosos e tem que manter o sincronismo

O **ALOHA(puro)** é mais simples, não tem sincronismo, quando um quadro chega ele já envia imediatamente, sem esperar o outro acabar e se houver colisão e vai porque as colisões aumenta muito, ele espera um tempo aleatório para tentar ser enviado novamente, quando um rede é menor e tem pouco nó esse protocolo é muito eficiente, mas à medida que os nó as colisões sobem quase que exponencialmente

revezamento

No protocolo de revezamento tempos 2 tipos o **polling** e o **token**. No **polling** um dos nó será o nome "mestre" e o outros os nós "filhos" o nó mestre fica verificando/chamando os nós "filhos" a fazerem a transmissão de dados, se o nó for fazer a transmissão o nó mestre concede um número de quadro para realizar a transmissão, quando esse nó terminar, passa para o próximo e depois pro próximo, e vai fazendo isso de forma circular, mas isso pode gerar uma sobrecarga, se o nó mestre falhar, cai o sistema todo.

No **token**, que é a permissão para o nó fazer a transmissão, esse token passa de forma circular por todos os nó, se o nó for fazer a transmissão e tiver com o token ele faz senão tem que esperar está com ele, uma analogia boa é a corrida em grupo, onde um atleta só pode correr se tiver com o bastão na mão, que assemelha ao token, mas neste protocolo temos que considerar a latência por ele ser circular e demorar muito até chegar de novo no primeiro nó e se o token falhar todo o sistema falha