**Jamkazam está usando um esquem patenteado para metrônomo distribuído.**

O vídeo do Jamkazam que fala sobre o metrônomo distribuído mostra muito bem que quando um baterista é a referência da banda a latência fica dobrada para o baterista. O baterista toca, o áudio demora 20 ms para chegar no outro músico. Esse músico toca junto com o baterista. Para o outro músico está tudo ótimo, mas para o baterista a latência será dobrada, o tempo necessário para o áudio do baterista chegar no outro músico somado com o tempo para o áudio do outro músico chegar no baterista. A ideia abaixo poderia reduzir a latência pela metade, o que é muito bom.

Pensei na possibilidade uma máquina atuar como máster. Quando o usuário máster clica em iniciar o metrônomo a máquina envia uma mensagem para os peers. Os peers vão receber essa mensagem com alguns ms de atraso em relação ao tempo do máster. O metrônomo dos peers inicia assim que eles recebem as mensagens. Esse metrônomo está atrasado rem relação ao máster, e como o áudio do peer vai demorar o mesmo tempo para chegar na máquina máster é necessário ajustar o metrônomo. Supondo que exista 20 ms de atraso no metrônomo do peers, esse atraso poderia ser compensado adiantando cada batida do metrônomo em alguns ms, isso poderia ser feito nas 8 primeiras batidas, por exemplo (poderia deixar um count in de 2 compassos por padrão). Depois de feita essa correção o metrônomo do peer estaria teoricamente tocando junto com o metrônomo do máster, e a latência entre eles seria somente a latência da rede.

**Outra Ideia para sincronizar:** quando cada máquina conecta eu poderia armazenar o tempo do server. Esse tempo seria o relógio global. Cada cliente armazena esse offset. Quando o máster envia a mensagem de início de metrônomo ele envia o timestamp. Esse timestamp seria o offset em relação ao relógio global. Quando o cliente recebe a mensagem ele deve calcular o momento exato do start usando o seu próprio offset em relação ao relógio global. Com isso teria como saber exatamente o momento em que o metrônomo iniciou no máster. Claro que o cliente estaria atraso por conta da latência da rede. Esse atraso tem que ser calculado e distribuído entre as 8 batidas do metrônomo da mesma forma como pensei na ideia anterior. De qualquer maneira eu vou precisar da latência da rede, e isso pode mudar. Então acho que uma ideia futura seria um metrônomo variável, se ajustando minimamente de acordo com a variação na latência da rede.

Supondo que existe uma diferença de 40 ms entre os peers. Com o metrônomo sincado o áudio chegará com 40 ms de atraso em cada máquina. Atrasando o áudio local em uns 15 a latência final ficará em 40-15 = 25 ms, ou seja, vale muito a pena atrasar o áudio local.

**Problema:** esqueci de um detalhe: quando me conecto com o servidor existe um tempo para a mensagem de conexão ir e voltar. Esse tempo é diferente para cada cliente, portanto isso precisaria ser considerado. Na prática isso pode ser uma fonte de problemas já que no momento da conexão o tráfego pode estar desfavorável gerando um tempo maior para conectar. A solução parece envolver um servidor de tempo mesmo.

**Solução:** posso considerar o máster como o servidor de tempo. Todos os clientes conseguem calcular o offset deles em relação ao máster. O máster manda o comando de início de metrônomo e manda o timestamp dele. Os clientes recebem esse time stamp e fazem o calculo...

Mas espera um pouco... Acho que não preciso desse timestamp. O máster manda o comando de início, o cliente recebe e inicia o metrônomo. A latência entre máster e cliente será compensada nos primeiros beats do metromo, independente do momento em que a mensagem de início de metrônomo chegou.

**Outra possibilidade**

Pensei em uma ferramenta para ensaios, totalmente focada nisso, nada de jams

Em uma ferramenta para ensaios a estrutura da música estaria previamente definida. **Com uma estrutura pre-definida fica tudo mais fácil.**

**Mesmo com uma estrutura pré-definida como ficaria a questão da sincronia?**