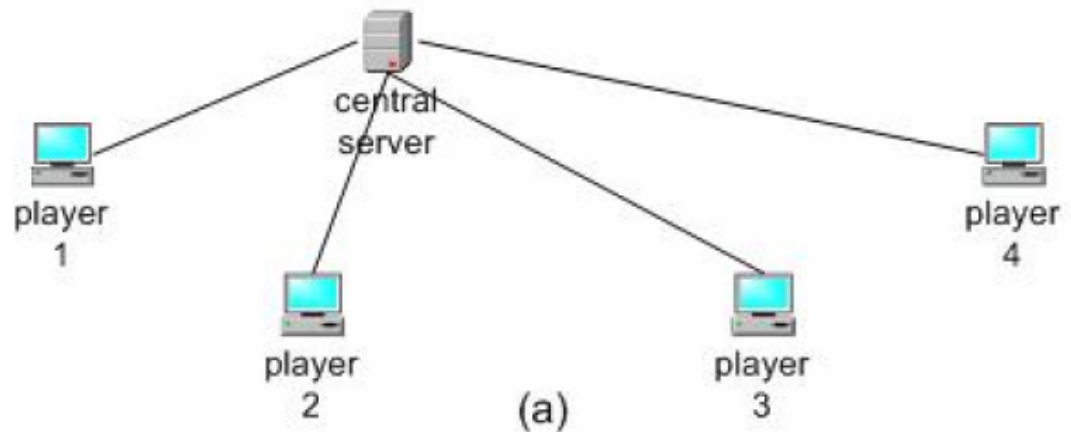


Jogos Multiplayer e Sistemas Distribuídos

- **Cliente-Servidor**

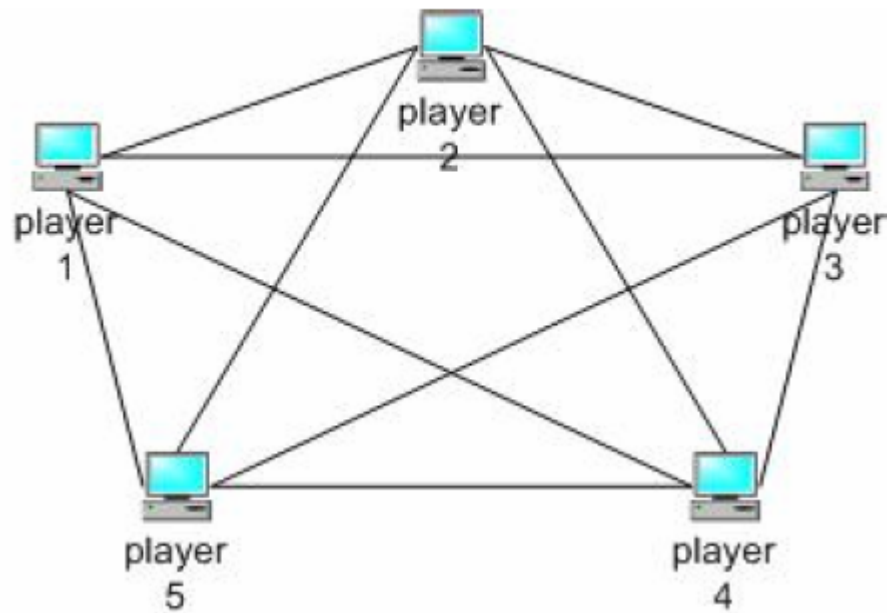
- Amplamente usada em design de Jogos Multiplayer Online (JMO);
- A maioria dos JMO's são formados na arquitetura Cliente-Servidor (*GuildWars*, *Everquest*), podendo suportar milhares de jogadores participando do mesmo jogo, ou um número mais limitado de jogadores (*Quake*, *Counter-Strike*);
- Todas as funções e informações são centralizadas em um único servidor com um grupo de Clientes conectados ao Servidor para enviar e receber dados;
- Como apenas o Servidor tem autoridade dentro de todo o jogo, não há problemas de consistência no sistema;
- Como todos os Clientes estão conectados a um Servidor, cada comando vai ter de ser enviado ao Servidor e depois enviado novamente para outros Clientes, adicionando **latência**;
- Como toda a comunicação entre Clientes têm que ser transferida pelo Servidor, a **carga de serviço** é **alta** e o Servidor se torna um **gargalo**.



- **Peer-to-Peer**

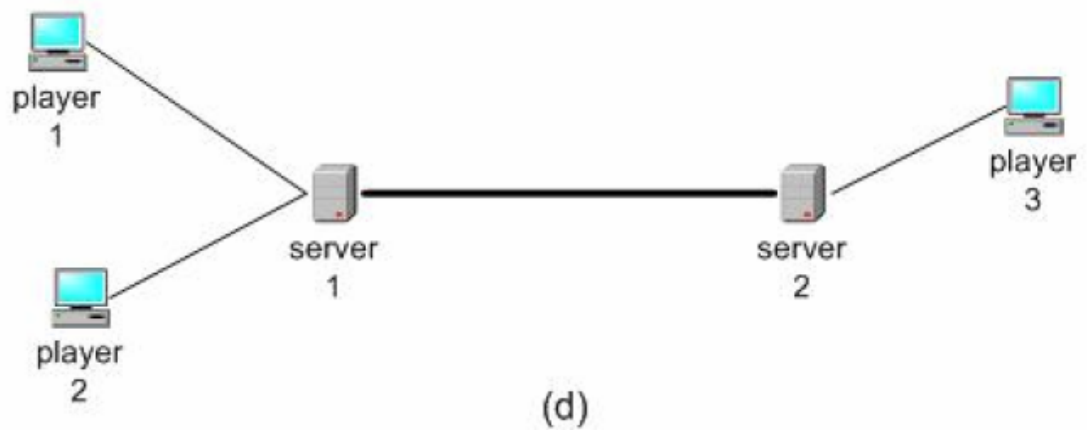
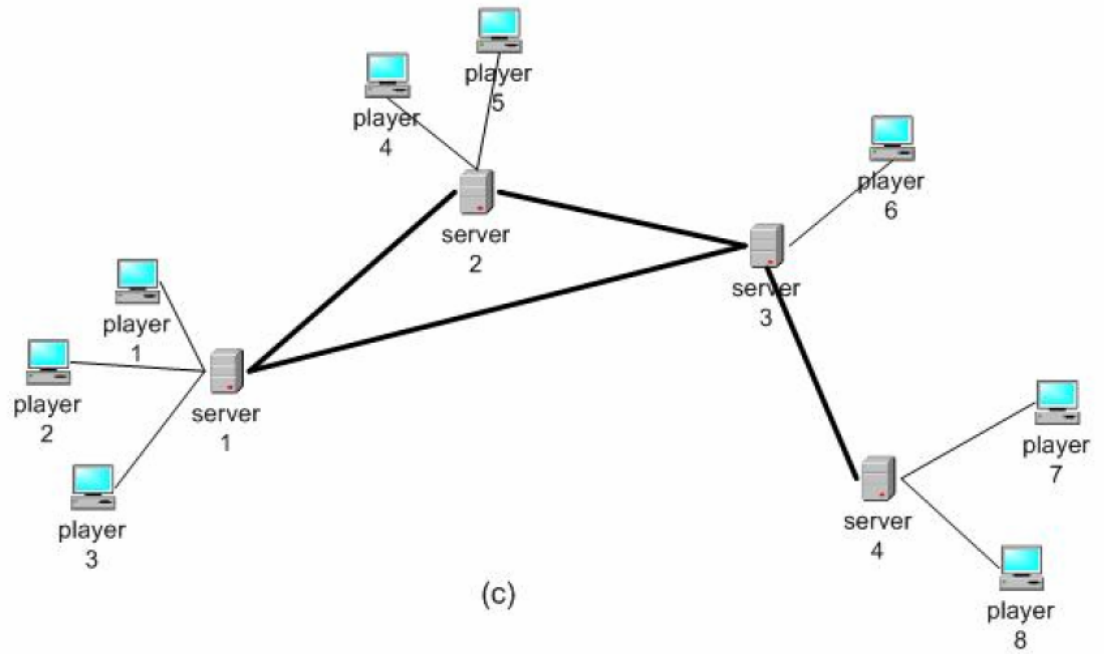
- Todos os Clientes se comunicam simetricamente entre si e possuem papéis iguais (*Starcraft*);
- Todo Cliente mantém um link com todo outro Cliente;
- Em teoria, Clientes em uma arquitetura peer-to-peer podem ter um delay menor comparado à arquitetura Cliente-Servidor;
- Sem um Server Central, não há um ponto de interrupção óbvio, **tolerância a falhas maior**;

- Esta arquitetura requer que todo Cliente possua uma cópia de todo o jogo, assim é necessário uma forma de **sincronização entre Clientes** para garantir que toda cópia do jogo esteja no mesmo estado;
- O número de conexões necessárias cresce rapidamente de acordo com que novos jogadores entram no jogo, sendo assim, em jogos MMORPG a arquitetura pode ter um custo alto se uma conexão de malha completa for necessária.



- **Servidor Distribuído**

- Sendo uma combinação das duas arquiteturas anteriores, uma arquitetura de Servidor Distribuído herda parte das vantagens das arquiteturas Cliente-Servidor e Peer-to-Peer;
- Como todos os Servidores estão distribuídos Geograficamente, todo Cliente pode escolher um Servidor que forneça a menor latência;
- O tempo de resposta de Clientes pode ser melhorado com uma arquitetura de Servidores Distribuídos;
- Uma estrutura multi-servidor distribui a carga do Servidor, além disso, esta estrutura também distribui o risco de falhas no Sistema.



- **Bucket Synchronization Mechanism**

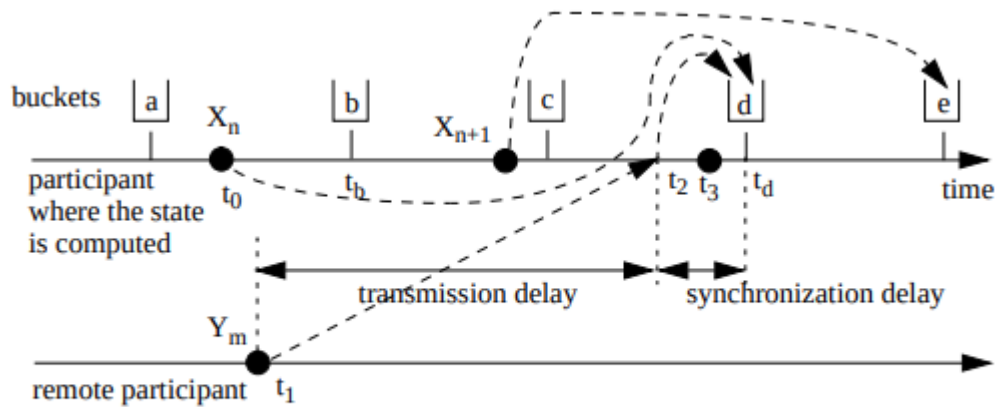


Figure 5. The bucket synchronization mechanism

- **Sincronização** foi introduzido para permitir que ações emitidas “ao mesmo tempo” serem processadas em conjunto por cada participante;
 -
- **Navegação estimada**