

Trabalho de conclusão de curso

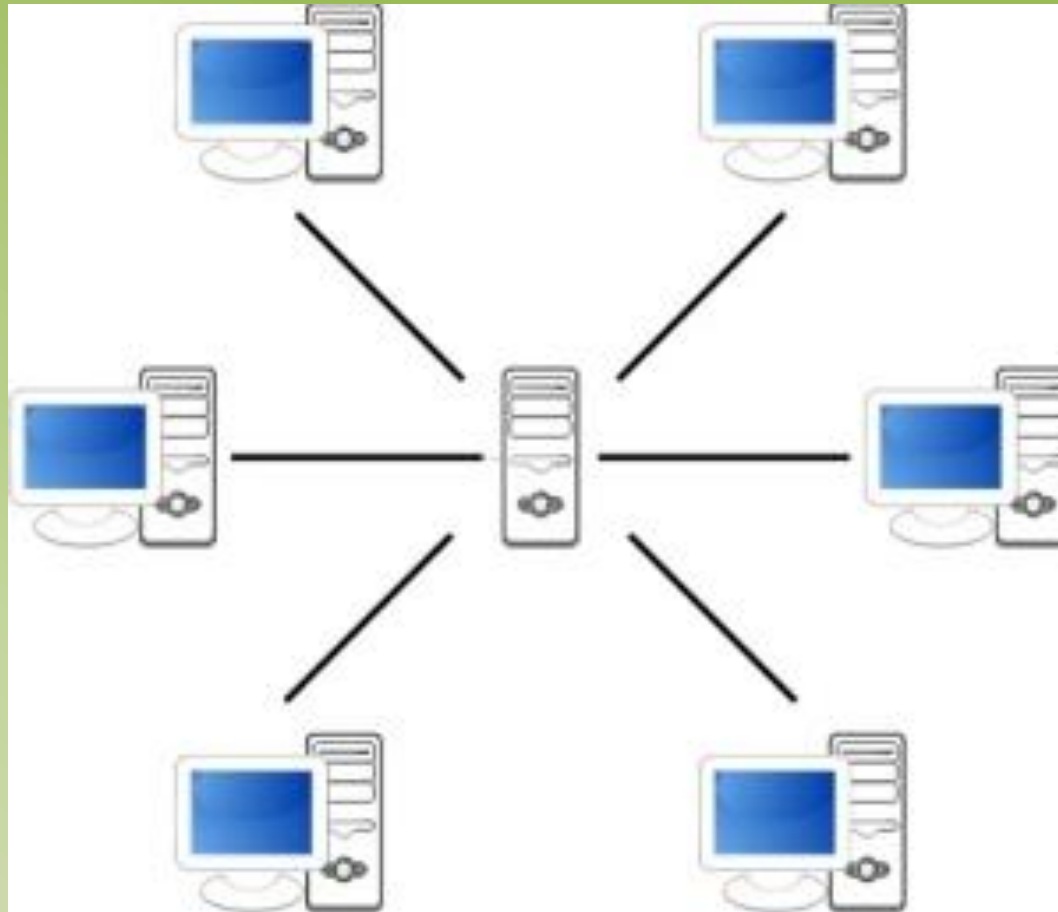
Desenvolvimento de um protocolo de comunicação P2P em rede local para jogos e implementação de um plugin para o motor de jogos Unity 3D

Por Renato Scaroni
Supervisão: Daniel Macêdo Batista

Comunicação de rede em jogos

- Em geral baseada em estrutura clientes-servidor:
 - Exige que um peer na rede seja dedicado ao papel de servidor
 - Todos os dados trocados entre jogadores passam pelo servidor

Comunicação de rede em jogos



Esquema de comunicação de rede clientes-servidor

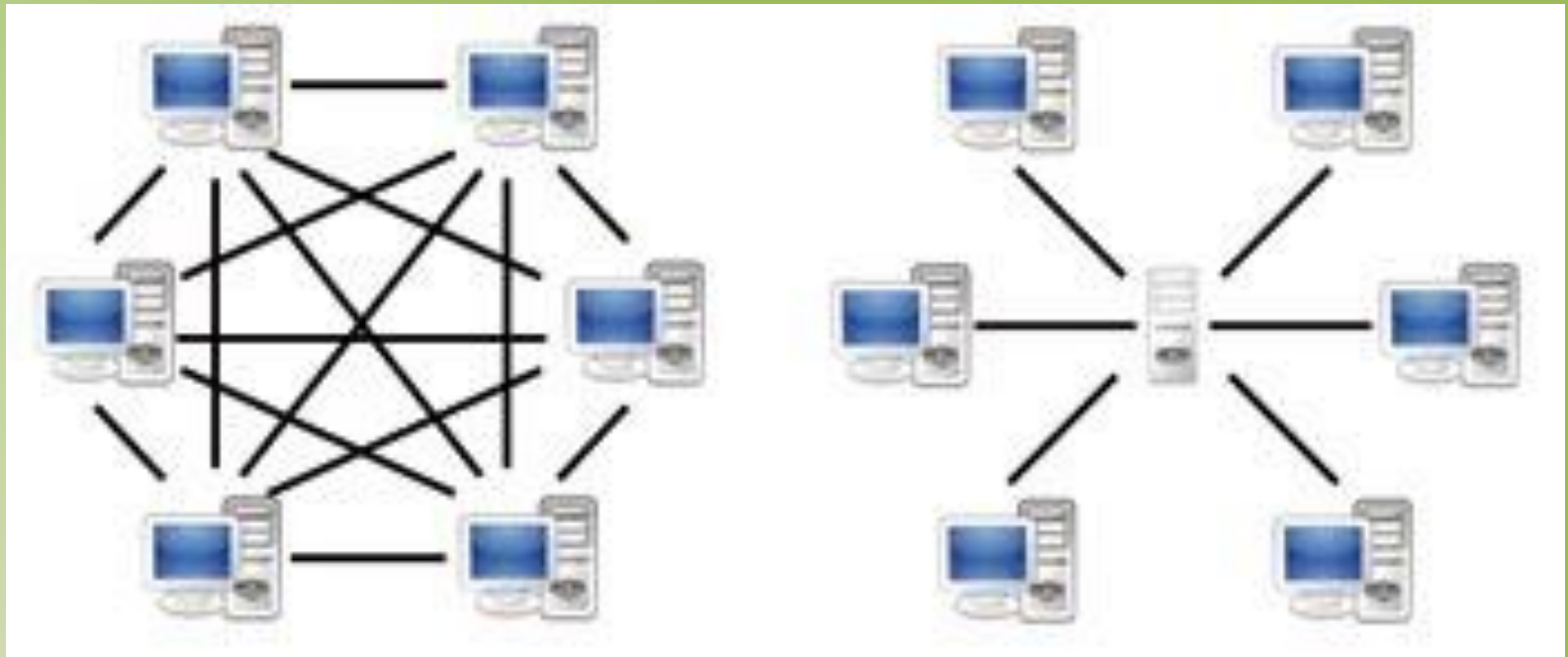
Comunicação de rede em jogos

- Problemas de uma rede clientes-servidor:
 - Todos os dados trocados entre jogadores passam pelo servidor (gargalo)
 - Exige que exista um servidor
 - Qualquer problema com o servidor implica que não haverá jogo
 - Inconveniente para jogos casuais em rede local

Comunicação de rede em jogos

- Comunicação em P2P
 - Peer to peer
 - Não existe distinção entre os papéis dos peers em rede
 - Todos os peers na rede são capazes de requisitar e prover dados aos demais da rede
 - Mais prático para criação de partidas rápidas entre os jogadores

Comunicação de rede em jogos



Esquema de comunicação de rede P2P X clientes-servidor

Comunicação de rede em jogos

- Problemas da comunicação em P2P
 - Descobrir jogadores na rede
 - Garantir disponibilidade da informação
 - Garantir eficiência na troca das mensagens
 - Minimizar atrasos e garantir que todos recebam as informações quando devem receber
 - Poucas opções de ferramentas prontas que implementem esse tipo de rede eficientemente
 - Desenvolver um sistema P2P não é uma tarefa trivial

Plataforma Utilizada



Motivação e objetivos

- Propor um protocolo de comunicação em rede local P2P
- Avaliar a viabilidade da implementação de um novo protocolo de comunicação P2P na Unity 3D
- Implementar o protocolo através de um plugin que integre com a Unity 3D

Protocolo proposto

- Ocorre em 3 fases:
 - Fase “Stand By”: na qual os jogadores escutam por novas conexões na rede
 - Fase de conexão individual: na qual os jogadores constroem as conexões de rede diretas entre cada um
 - Fase de jogo: Os jogadores trocam mensagens entre si através das conexões criadas

Protocolo proposto

- Fase “Stand by”:
 - Jogadores informam sua existência a partir de mensagens de *alive*.
 - *Broadcast* em rede local.
 - Qualquer peer conectado à rede saberá da existência dos demais.

Protocolo proposto

- Fase de conexão individual:
 - A partir de uma lista dos jogadores que forem descobertos na fase anterior são montadas conexões diretas entre cada peer
 - Canais diretos para evitar a sobrecarga de pacotes gerada pelo broadcast (possível causa para perda de pacotes)
 - Conexões são feitas utilizando sockets UDP
 - Garante agilidade na comunicação

Protocolo proposto

- Fase de jogo ou de troca de mensagens:
 - Cada peer envia uma mensagem a cada número fixo de frames que renderiza
 - A cada mensagem enviada temos uma marcação de tempo (timestamp)
 - Uma mensagem atrasada não é aceita como mensagem válida
 - Garante que a informação não é antiga

Implementação

- Por que Unity?
 - Uma das mais utilizadas por desenvolvedores indie
 - Multiplataforma
 - Não possui suporte a redes P2P na implementação atual do módulo de redes
 - Pouquíssimas alternativas de rede

Implementação

- Integração com a API da Unity:
 - Uso de eventos para chamar trechos do código que não precisam ser MonoBehaviour ou que rodam assincronamente
 - Fácil integração com aplicações em geral
 - Basta injetar um script em um objeto e pronto!

Conclusões e trabalhos futuros

- O que foi feito:
 - Desenvolvimento do protocolo
 - Teste de viabilidade e implementação do plugin
 - <http://github.com/renato-scaroni/mac499>

Conclusões e trabalhos futuros

- Desafios:
 - Entender e lidar com problemas não previstos em relação ao protocolo P2P (desenvolver um protocolo de fato não é algo trivial)
 - Algumas incompatibilidades da Unity em relação a alguns métodos na classe de sockets do .Net (em especial quando envolve threads e comunicação assíncrona)

Conclusões e trabalhos futuros

- Pontos positivos:
 - Ver vários conceitos vistos em áreas e disciplinas diferentes sendo usados em conjunto em um único trabalho
 - Juntar dois tópicos sobre os quais muito me interesse em um só trabalho

Conclusões e trabalhos futuros

- Trabalhos futuros:
 - Depurar o que já foi feito
 - Integrar com o Jogo Shuttles

Perguntas?

