Suporte distribuído a MMOGs em cenários com recursos limitados

Carlos Eduardo B. Bezerra
Orientador: Prof. Dr. Cláudio F. R. Geyer
Porto Alegre, RS, 25/11/2008

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Informática UFRGS

GPPD

Agenda

- Introdução
  - MMOGs
  - Modelo cliente-servidor
  - Modelo peer-to-peer
  - Exemplos de trabalhos relacionados
- O Modelo
  - Visão geral
  - Gerenciamento de interesse
  - Balanceamento visando a *hotspots*
- Trabalhos futuros
- Conclusão

Introdução:

Jogos maciçamente multijogador, seus requisitos e abordagens

Introdução: MMOGs

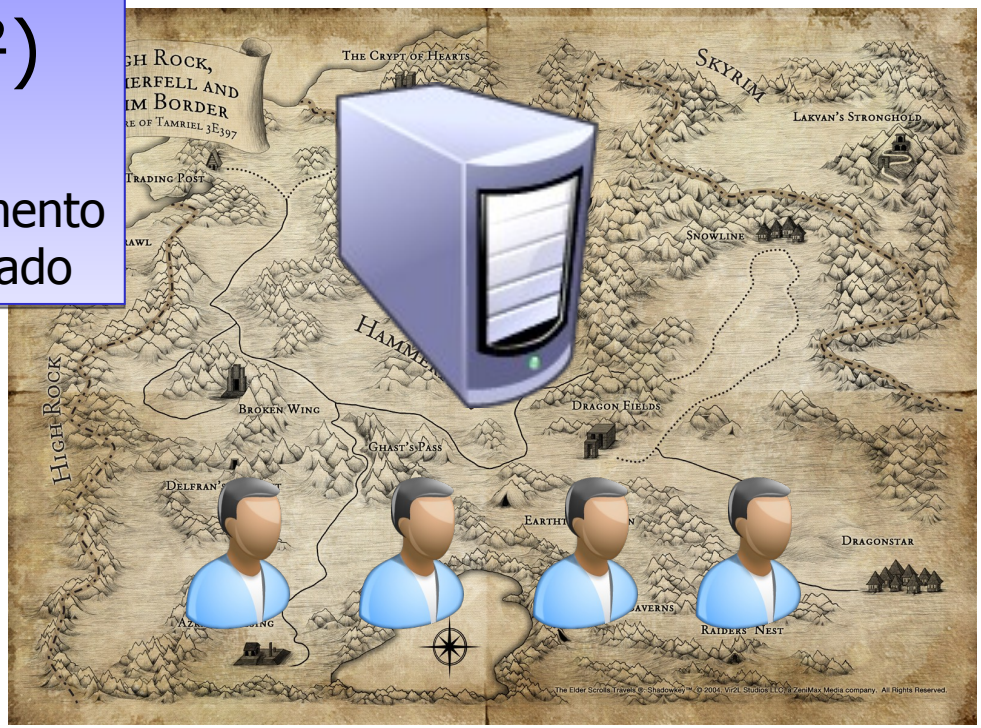
- MMOGs
  - *Massively multiplayer online games*, ou Jogos Online Maciçamente Multijogador
  - Grande número de jogadores simultâneos
  - Jogadores conectados através da Internet
  - World of Warcraft, EVE Online e EverQuest
  - Geram grande receita, porém implicam em grandes custos

- Interação dos jogadores
  - O ambiente do jogo é composto por **Entidades**
  - Cada jogador controla um **Avatar**, que tem diversos atributos
  - A interação ocorre através do envio de **Ações** e recebimento de atualizações de estado das entidades relevantes ao avatar
  - Problema: número de atualizações de estado (recebimento e envio) pode crescer **quadraticamente** em relação ao número de jogadores

Introdução: abordagem cliente-servidor

Tráfego
$O(n^2)$

Processamento centralizado

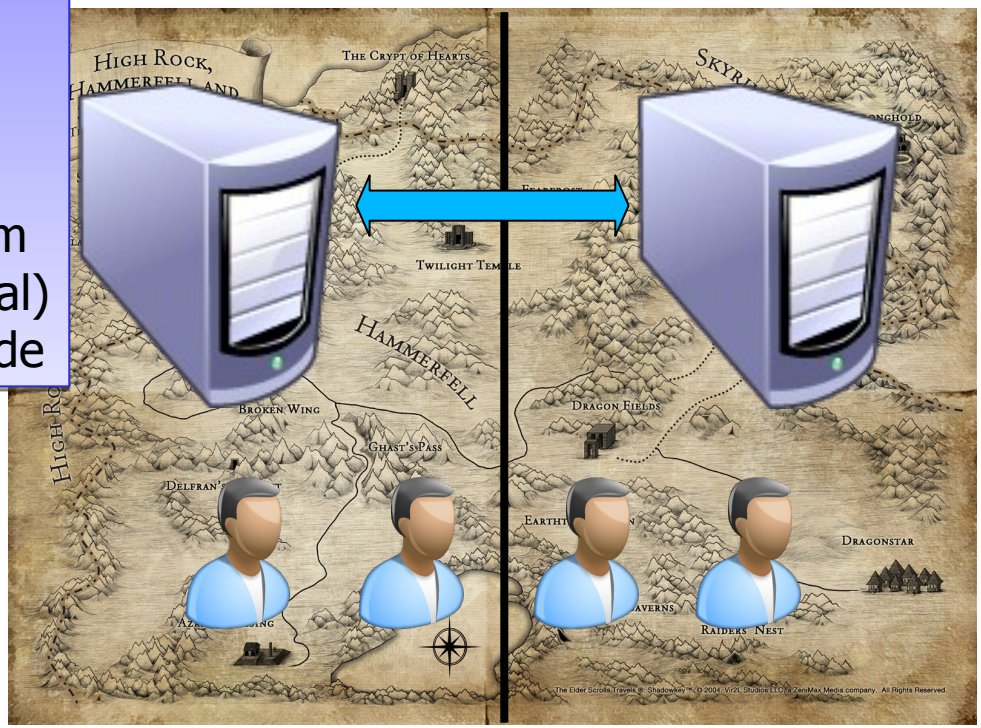


- Geralmente se usa a abordagem cliente-servidor
- O servidor recebe as ações, processa-as e envia os resultados
- Projeto mais simples que P2P, resistente a trapaça e permite controle central do jogo
- Escalabilidade é tratada com super-dimensionamento dos recursos

Introdução: abordagem cliente-servidor

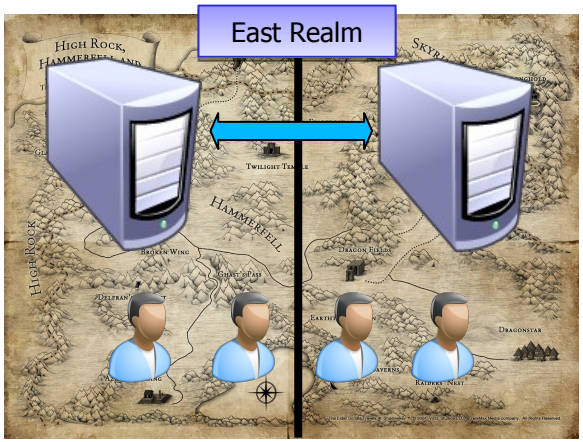
Tráfego
$O(n^2)$

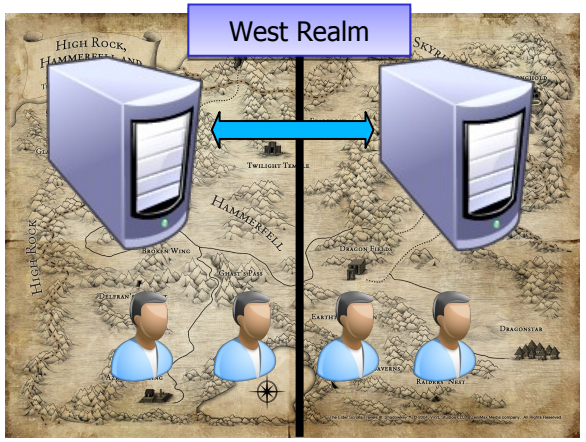
Processamento distribuído, porém em uma rede (local) com alta velocidade

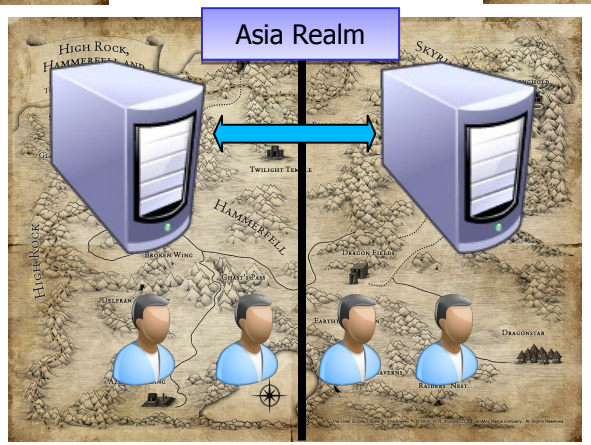


- Geralmente se usa a abordagem cliente-servidor
- O servidor recebe as ações, processa-as e envia os resultados
- Projeto mais simples que P2P, resistente a trapaça e permite controle central do jogo
- Escalabilidade é tratada com super-dimensionamento dos recursos

Introdução: sistemas servidores independentes







Tráfego
$O((n/R)^2) =$
$O(n^2/R^2)$

- Geralmente se usa a abordagem cliente-servidor
- O servidor recebe as ações, processa-as e envia os resultados
- Projeto mais simples que P2P, resistente a trapaça e permite controle central do jogo
- Escalabilidade é tratada com super-dimensionamento dos recursos

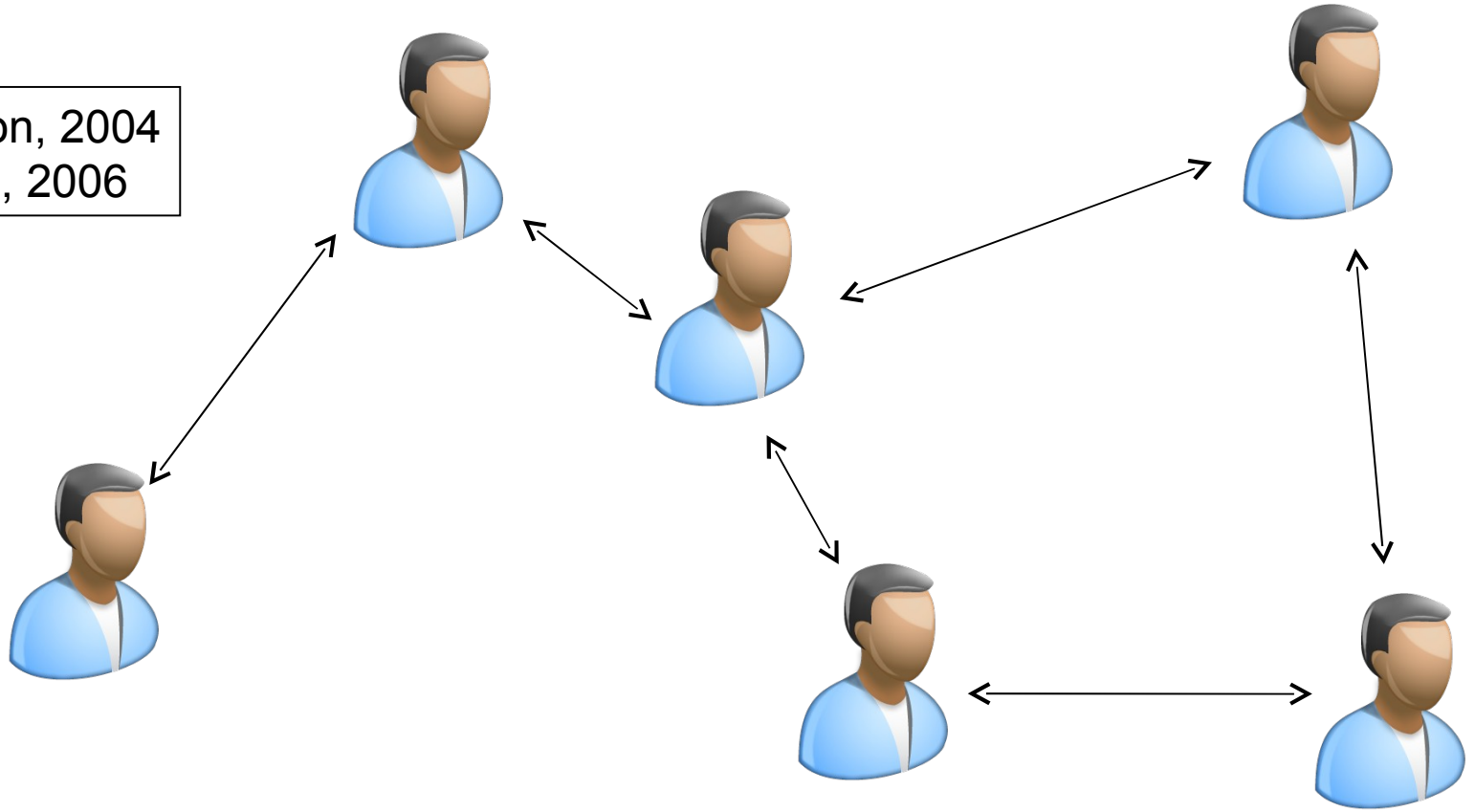
Introdução: abordagem P2P

- Uma alternativa comumente proposta é a P2P
  - Descentralização do suporte ao jogo
  - Eliminação do *single point of failure*
  - Mensagens trocadas diretamente implicam em menor atraso
  - Seria o ideal, não fossem algumas questões críticas

- Simulação executada pela máquina do jogador
  - Vulnerabilidade a trapaça
  - Possibilidade de inconsistência no estado do jogo
  - Sobrecarga da banda de upload dos jogadores

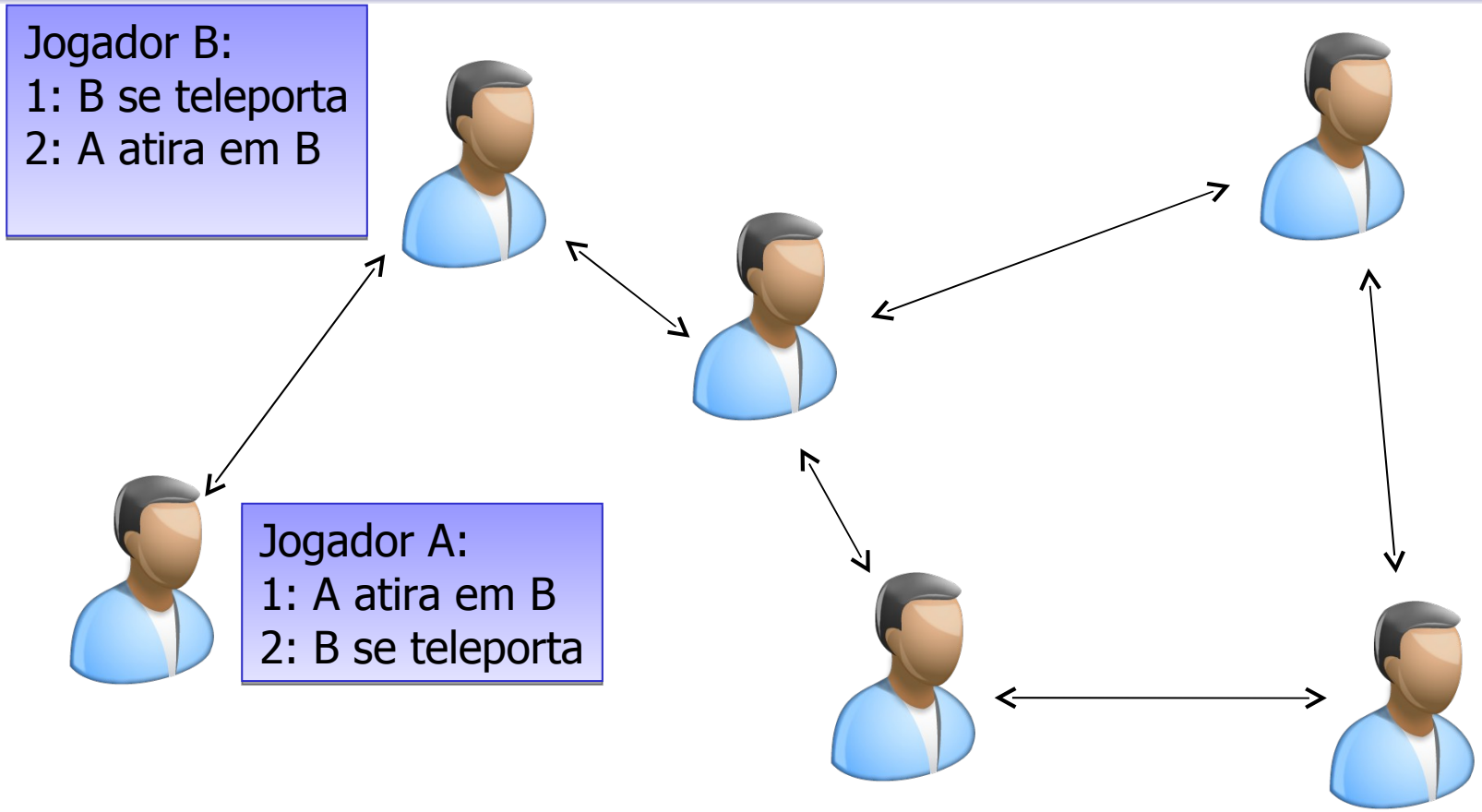
Introdução: questões em P2P (1/5)

Knutsson, 2004
Hampel, 2006



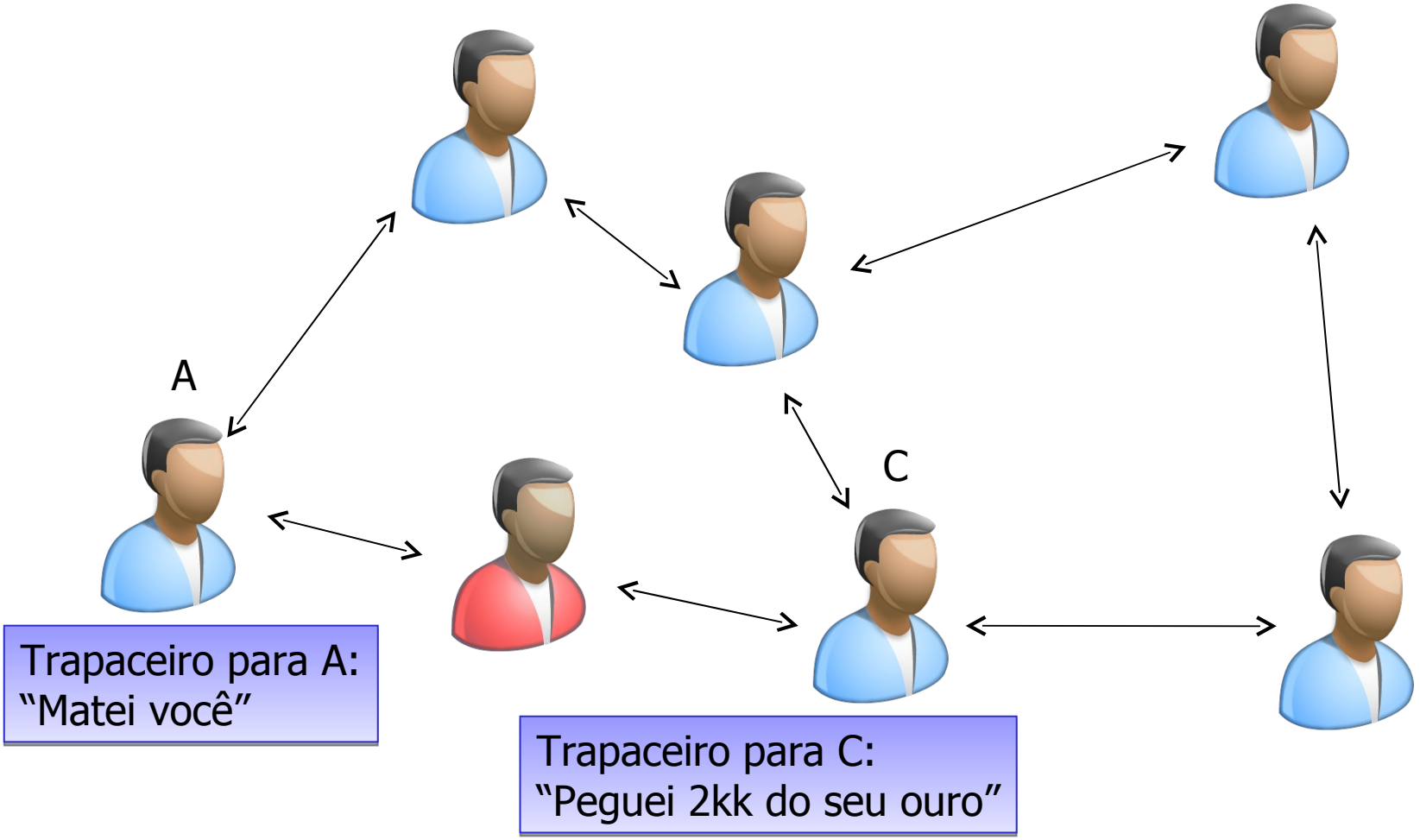
Cada jogador é responsável por simular e atualizar os outros

Introdução: questões em P2P (2/5)



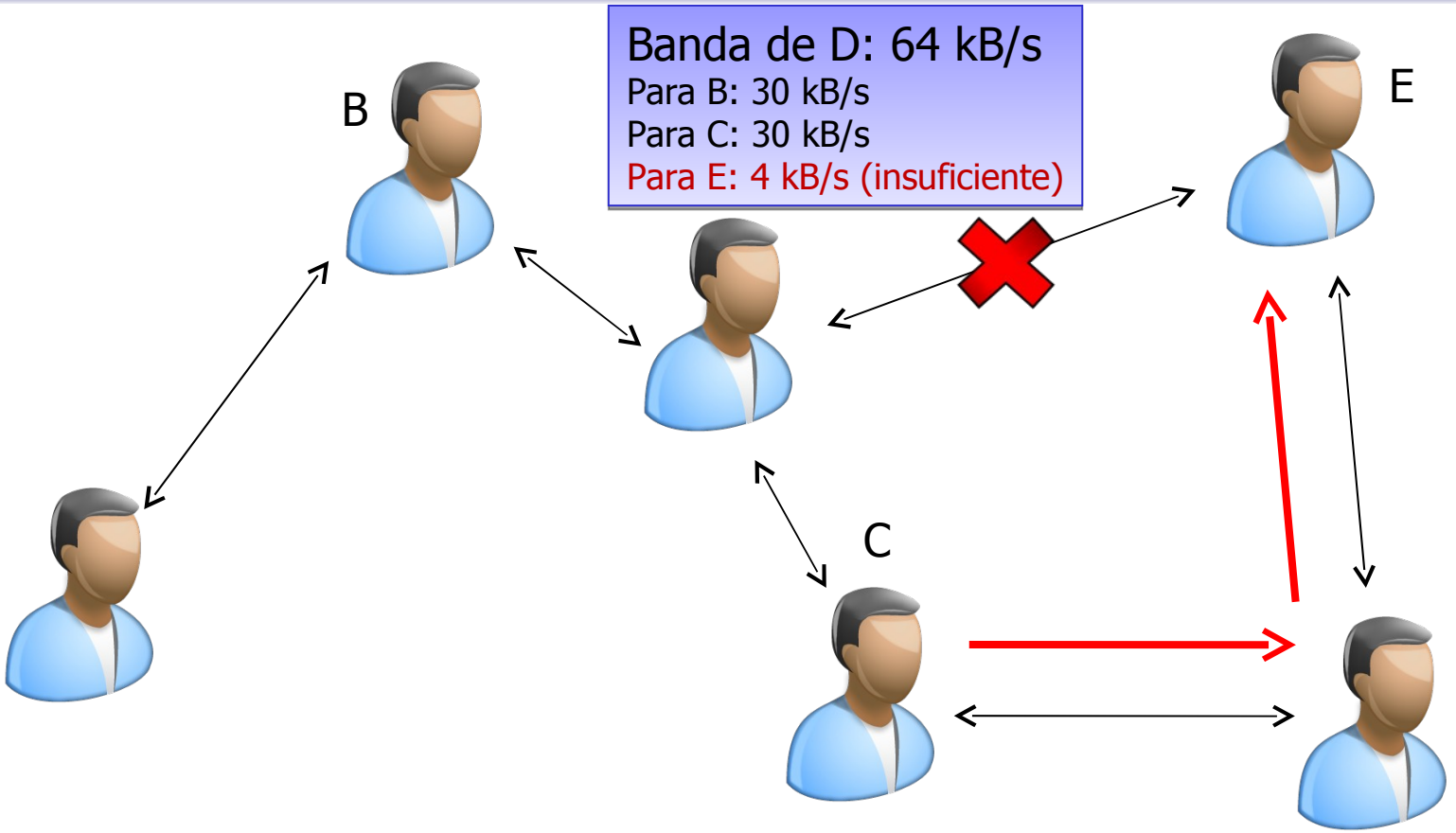
Pode haver **inconsistência** nas simulações

Introdução: questões em P2P (3/5)



Vulnerabilidade a **trapaça** sem árbitro central

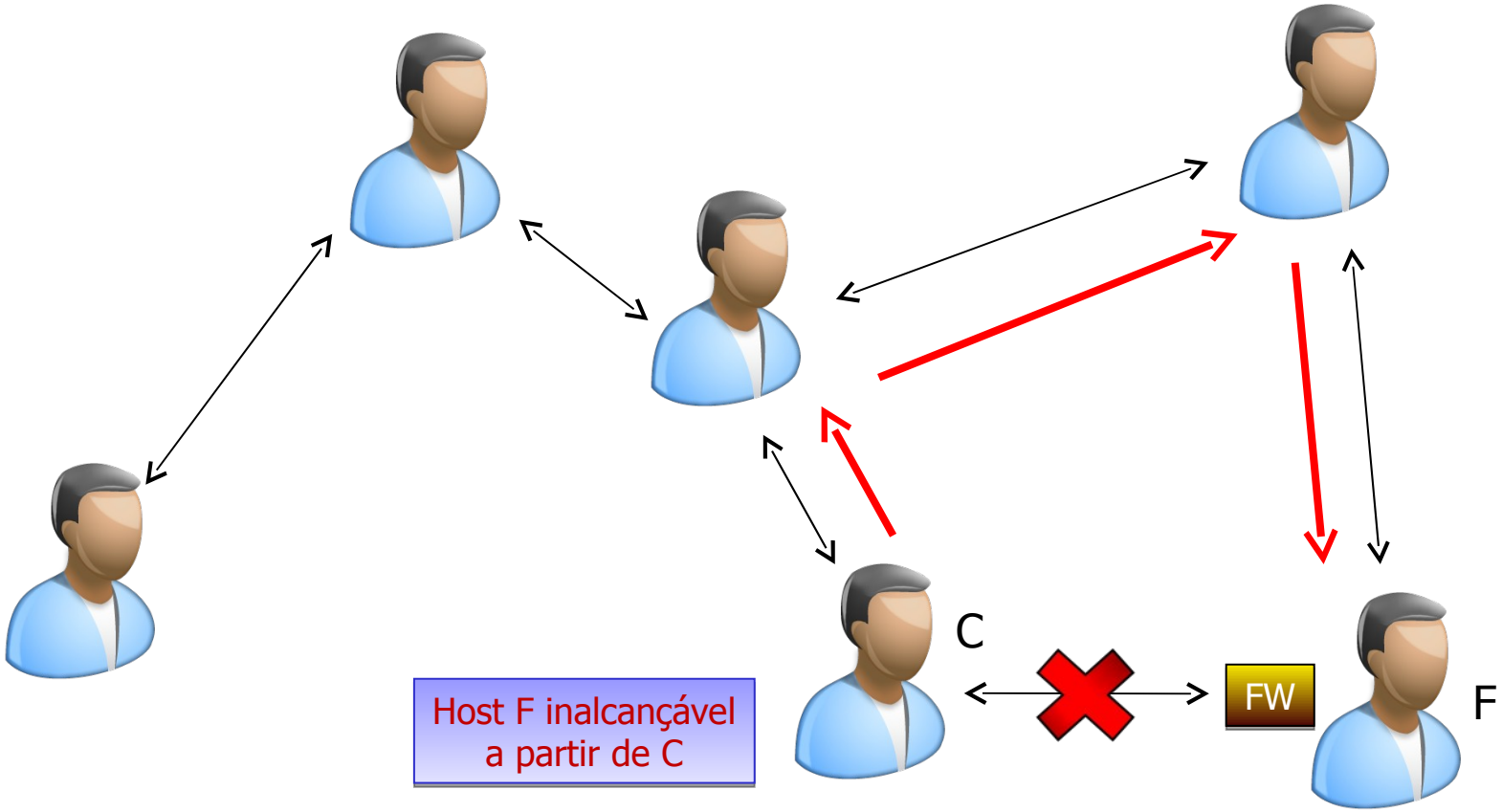
Introdução: questões em P2P (4/5)



Provável **sobrecarga** da banda de upload de algum peer

Pode ser feito roteamento, delegando os updates de um jogador a outro

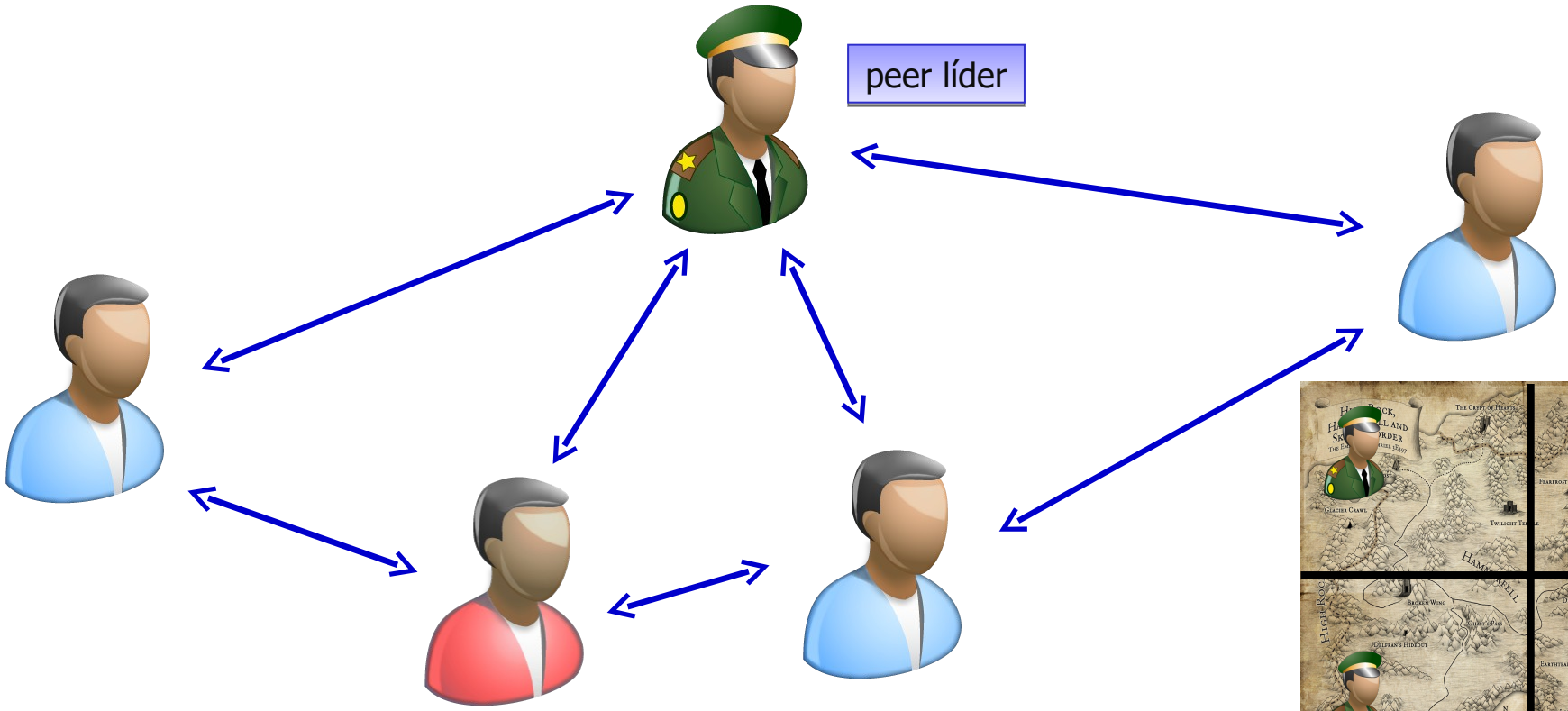
Introdução: questões em P2P (5/5)



Possível bloqueio por **NAT/Firewall**

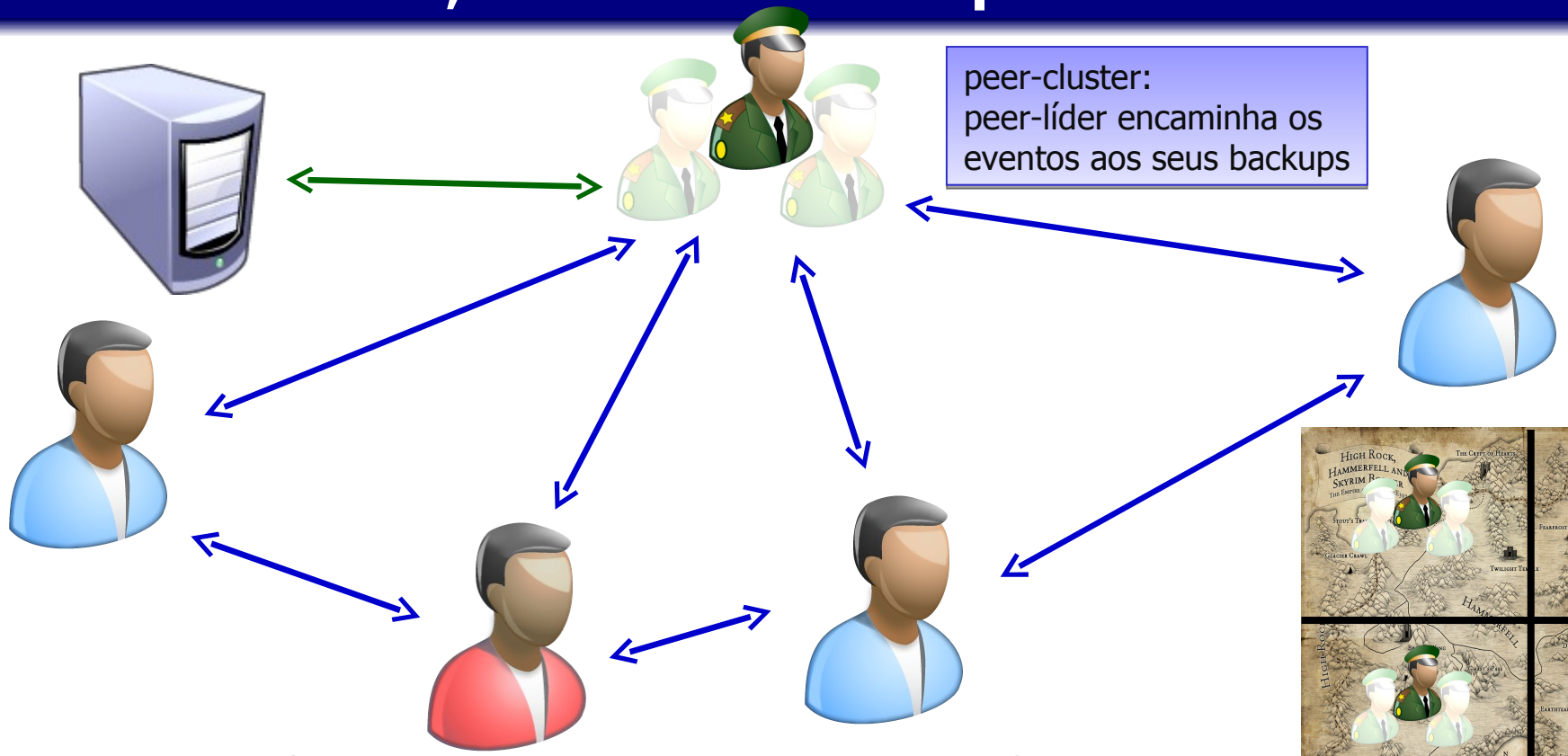
Necessário executar algum **roteamento**, desviando do bloqueio no destino

Hampel, 2006: eleição de um líder



+ Ordenação dos eventos do jogo
+ Verificação de trapaça
+ Broadcast de mensagens de peers com conexão mais lenta
– Possível sobrecarga de cpu e rede do líder
– Necessidade de o líder ser confiável

Chen, 2006: servidor e peer-cluster



+ Ordenação dos eventos do jogo e verificação de trapaça
+ Broadcast de mensagens de peers com conexão mais lenta
+ Tolerância a falhas do líder
+ Simulação do peer líder é verificadao pelos seus backups
– Sobrecarga ainda maior de cpu e rede do líder (pode recorrer ao servidor)
– Ainda sujeito a trapaça do peer-líder

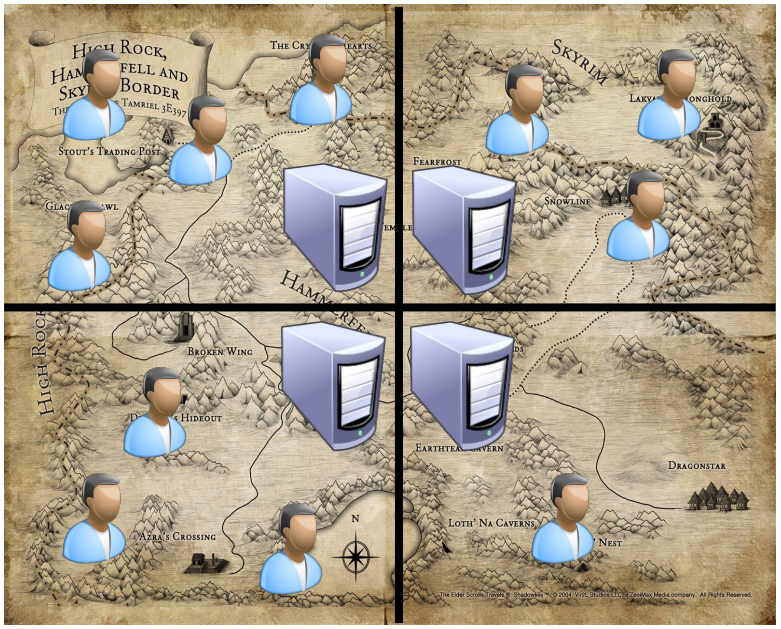
O modelo proposto

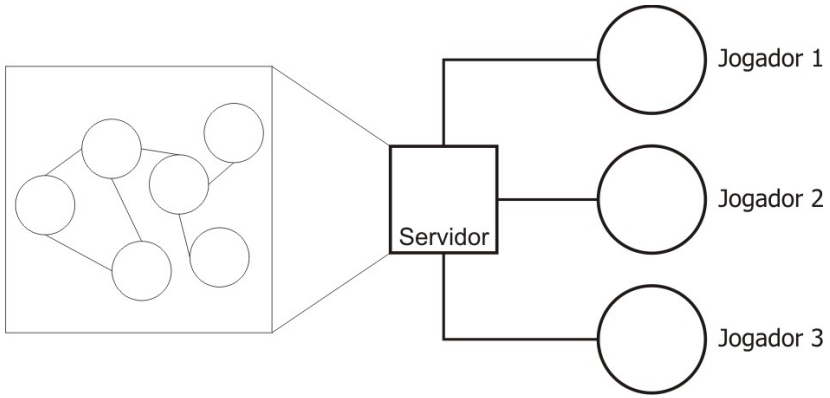
Visão geral: Objetivo

- Objetivo:
  - Prover um suporte para MMOGs de menor custo, mantendo segurança, consistência, escalabilidade e resistência a trapaça

- Proposta:
  - Prover um suporte para MMOGs, através da formação de um sistema geograficamente distribuído de nodos servidores ligados através da Internet, utilizando técnicas que reduzam o quanto for possível o tráfego do jogo, em detrimento da carga de processamento

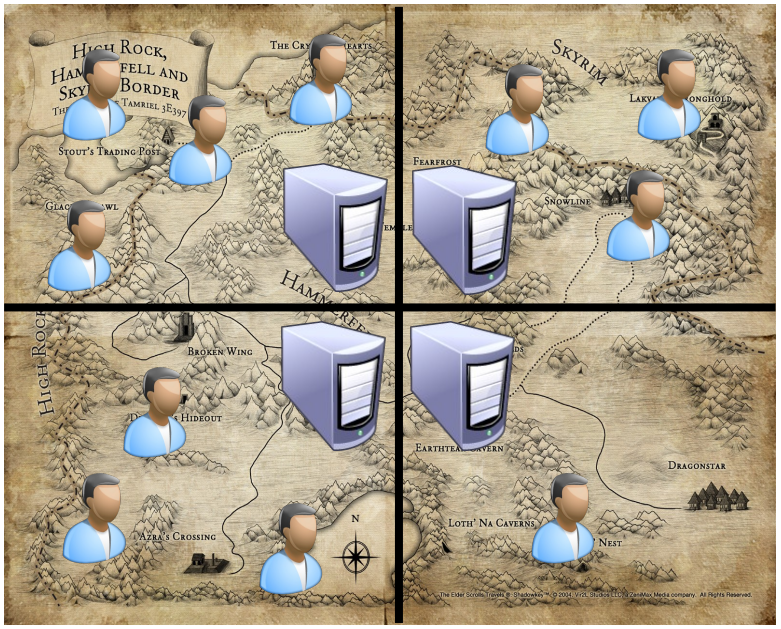
Visão geral: Modelo de distribuição

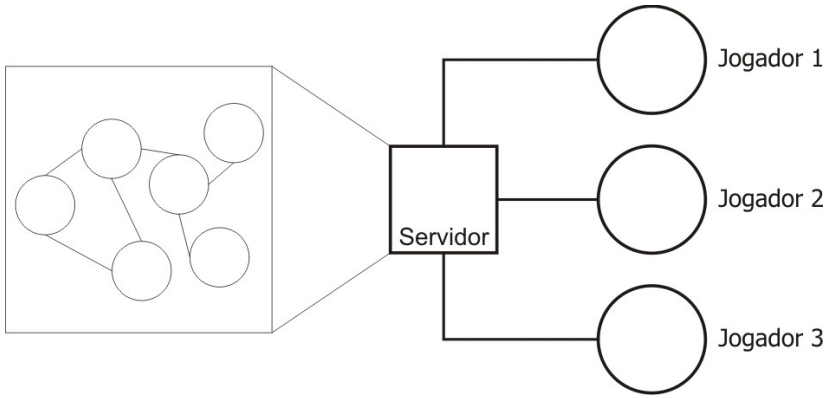




- A cada região é associada a um nodo servidor:
- Servidores em regiões vizinhas são adjacentes na rede overlay
- Cada jogador conecta-se ao servidor da região onde está situado seu avatar
  - Explora a localidade dos avatares
  - Servidores se comunicam para possibilitar interação entre jogadores em diferentes regiões
  - Mantém uma conexão já negociada com os servidores adjacentes ao primeiro (de regiões vizinhas)

Visão geral: Modelo de distribuição



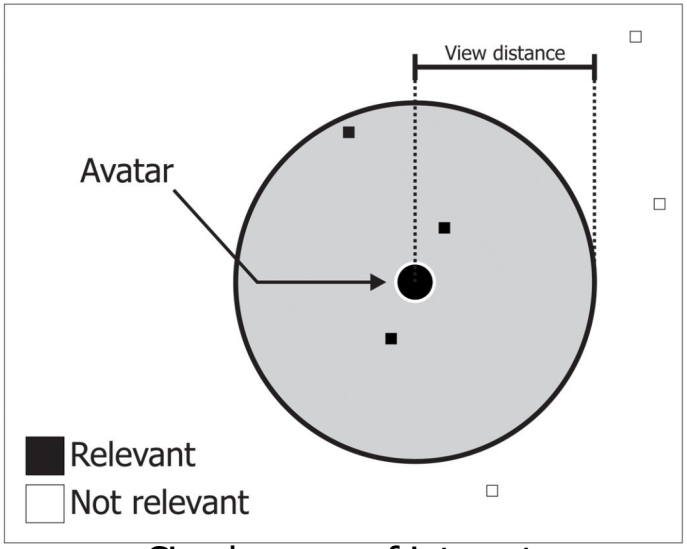


- Menor sobrecarga sobre os links dos peers → maior público para o jogo
- Servidores de baixo custo:
  - Como os servidores têm recursos limitados, é necessário otimizar o uso de banda, utilizando técnicas para tal fim, tais como: **gerenciamento de interesse** e **balanceamento de carga** com **detecção de hot-spots**

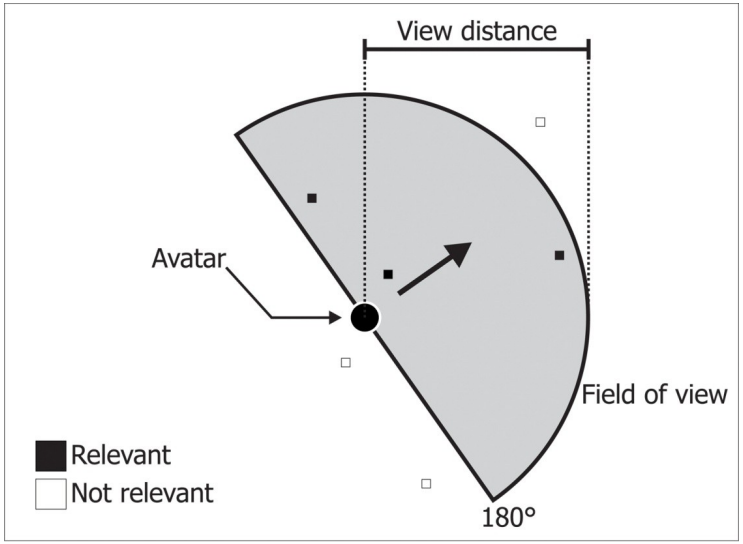
Gerenciamento de Interesse

Gerenciamento de Interesse

- Objetivo principal: reduzir o uso de largura de banda, filtrando o que é relevante ou não para cada jogador
- O cliente passa a ter um estado incompleto do jogo, mas isto não é perceptível
- Tráfego no servidor e clientes é reduzido
  - Servidores comportarão mais jogadores
  - Clientes terão menor requisito de banda disponível
- É necessário calcular o interesse de cada jogador, em relação a todas as entidades presentes no jogo
  - Geralmente com base na distância



Circular area of interest



Field of view based area of interest

(about 50% of the bw f the circular area)

Carlos Eduardo Benevides Bezerra

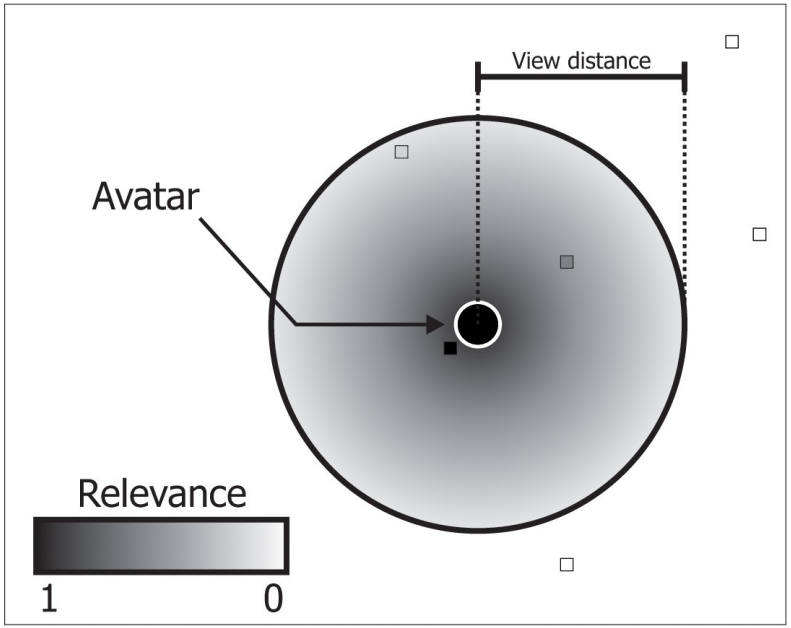
(A³) AOI with close Area and Attenuated update frequency

- Problema:
  - Consideram-se apenas duas possibilidades: relevante ou irrelevante
- Pode haver diferentes graus de relevância de uma entidade para um jogador
  - Como determinar o valor da relevância?
  - Como explorar diferentes relevâncias para ter economia de banda?

(A³) AOI with close Area and Attenuated update frequency

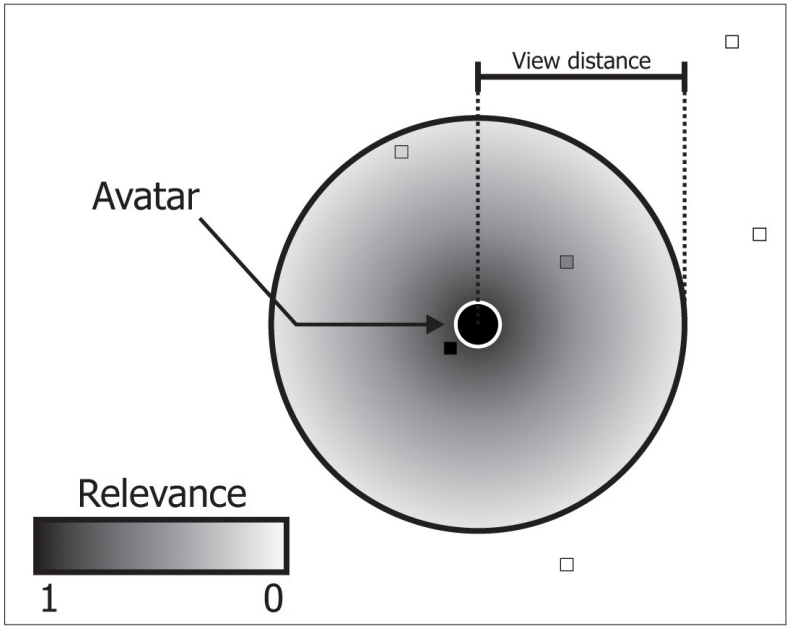
- Problema:
  - Consideram-se apenas duas possibilidades: relevante ou irrelevante
- Pode haver diferentes graus de relevância de uma entidade para um jogador
  - Como determinar o valor da relevância?
  - Como explorar diferentes relevâncias para ter economia de banda?

- Pode-se considerar que quanto mais distante, menos importante é um objeto
- Quanto mais importante é um objeto, mais frequentes são suas atualizações de estado
- Quanto menos importante, mais lento, até não receber mais nenhum update



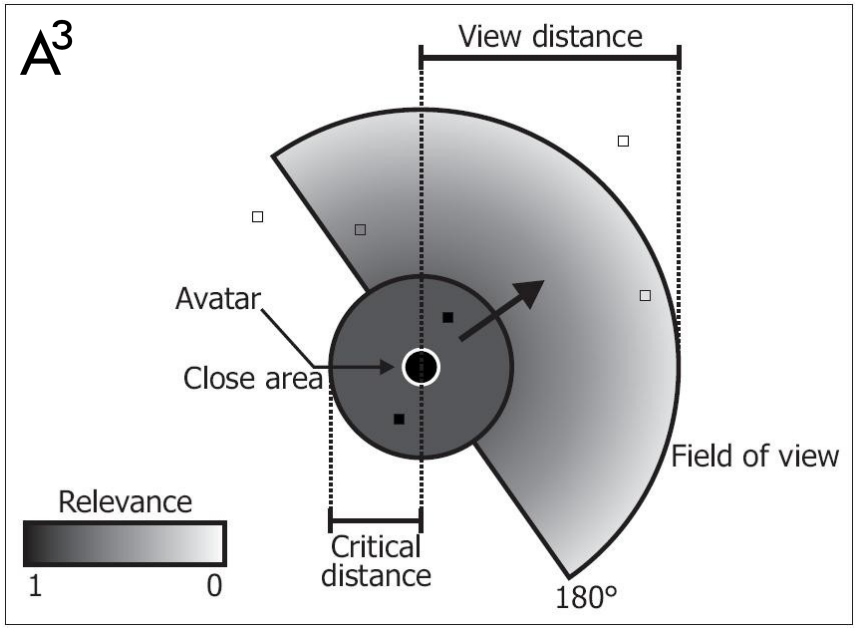
(A³) AOI with close Area and Attenuated update frequency

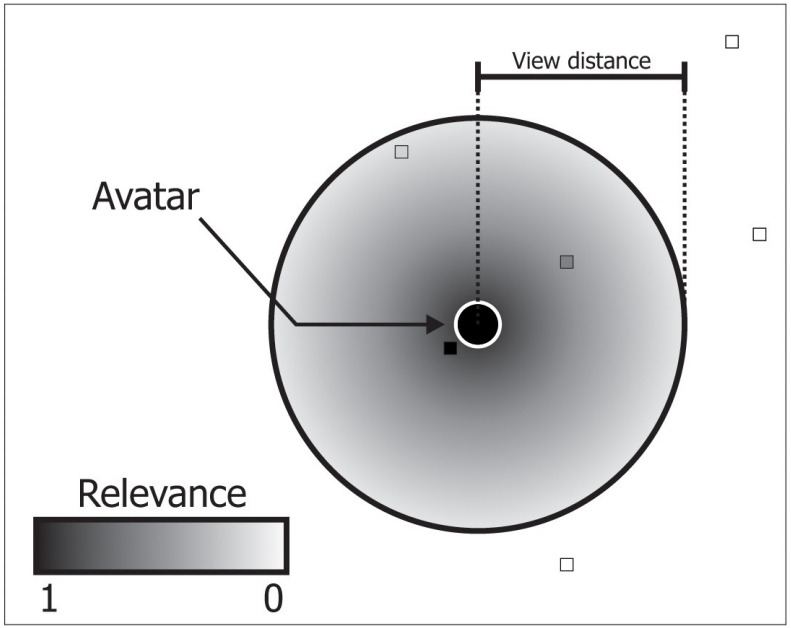
- Agora, há infinitos graus de relevância
- É possível refinar ainda mais o esquema de gerenciamento de interesse...



(A³) AOI with close Area and Attenuated update frequency

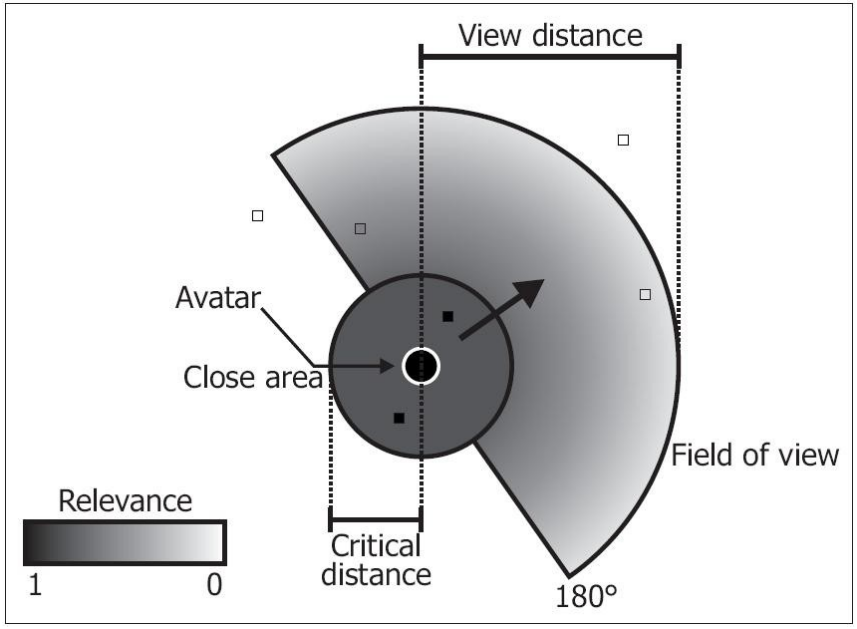
- Agora, há infinitos graus de relevância
- É possível refinar ainda mais o esquema de gerenciamento de interesse:
  - Pode-se usar o conhecimento do vetor direção do avatar e calcular seu campo de visão
  - Porém, é necessário levar em consideração entidades atrás do avatar





Algoritmo A³ - parâmetros

- Intervalo normal de atualização (I_N)
  - Menor intervalo de tempo entre duas atualizações de estado consecutivas de uma entidade
- Alcance de visão (V)
  - Distância a até qual uma entidade pode estar de um avatar para ser visível por ele
- Distância crítica (C)
  - Distância dentro da qual qualquer entidade tem relevância máxima para o avatar
- Coordenadas do avatar (A) e da entidade (E) em questão



Algoritmo A³ - retorno

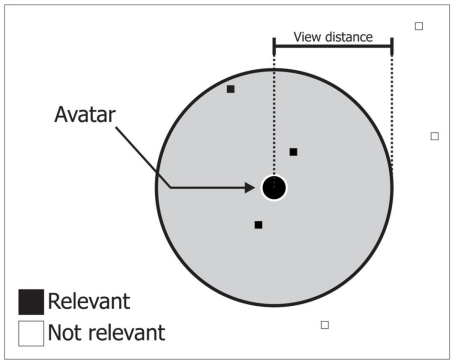
- Relevância (R)
  - Valor real x, $x \in [0, 1]$. Representa o quanto uma certa entidade E é relevante para o avatar A.
- Intervalo até a próxima atualização (I_P):
  - $I_P = I_N / R$
- Exemplo:
  - I_N = 200 ms
  - Distância implica em R = 0,8
  - I_P = 200 / 0,8 = 250 ms

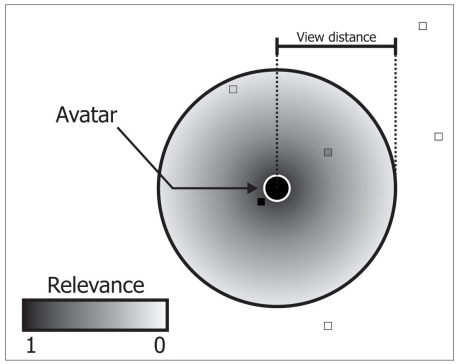
A³ - Algoritmo

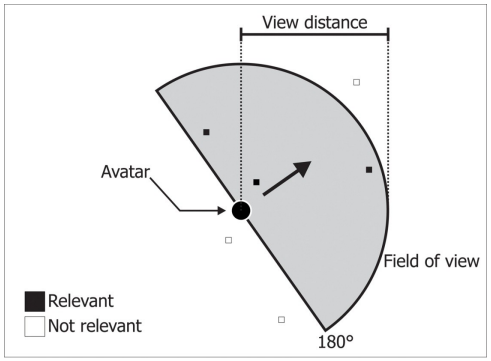
Algorithm 1 Calcular relevância da entidade E para o avatar A

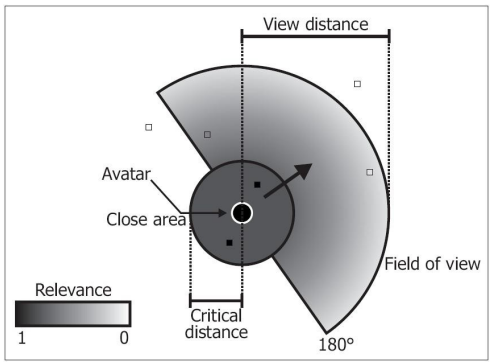
$dist \leftarrow distância(A, E)$
if $dist \leq distância\_crítica$ **then**
  $relevância \leftarrow 1$
else
  if A tem E em seu campo de visão **then**
    $relevância \leftarrow 1 - \frac{dist - distância\_crítica}{alcance\_da\_visão - distância\_crítica}$
    if $relevância < 0$ **then**
      $relevância \leftarrow 0$
    end if
  else
    $relevância \leftarrow 0$
  end if
end if

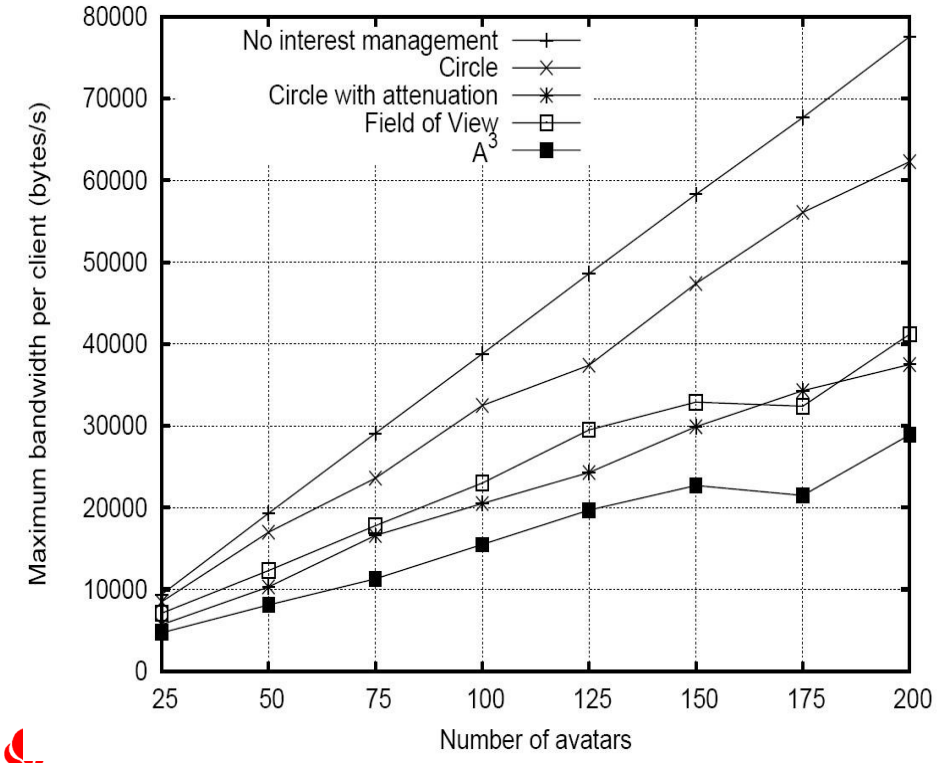
A³ - resultados

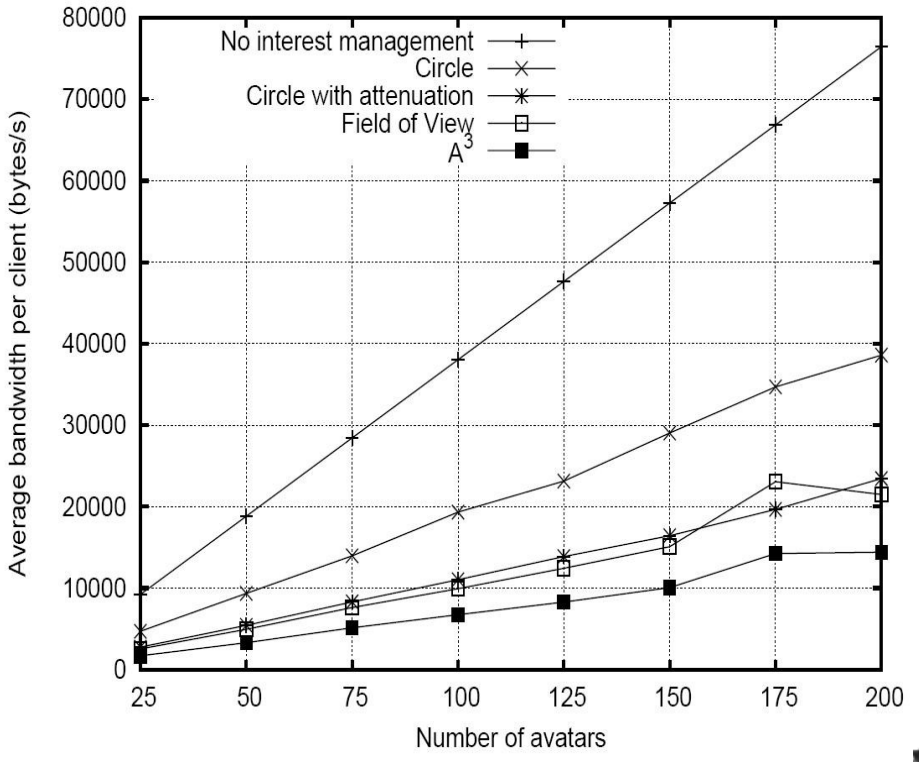












Balanceamento de Carga

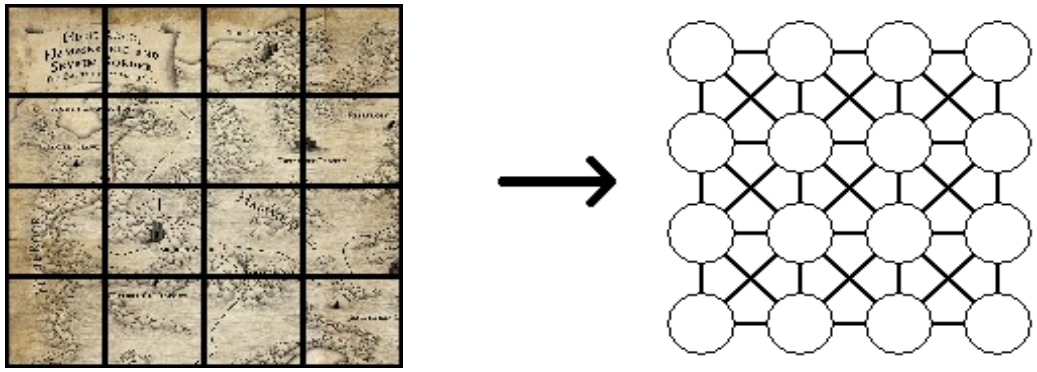
Detecção e gerenciamento de hotspots

Balanceamento de carga

- Pretende-se usar servidores de baixo custo formando um sistema heterogêneo
  - Servidores terão capacidade diferente uns dos outros
- Avatares dificilmente se distribuem de maneira uniforme no mundo
  - Hotspot: lugar de interesse comum, atraindo mais jogadores, criando um aglomerado destes
- Avatares de diferentes regiões têm interação com maior delay e maior custo de comunicação
  - Deve-se evitar ao máximo dividir um hotspot entre regiões

Balanceamento de carga – Microcélulas

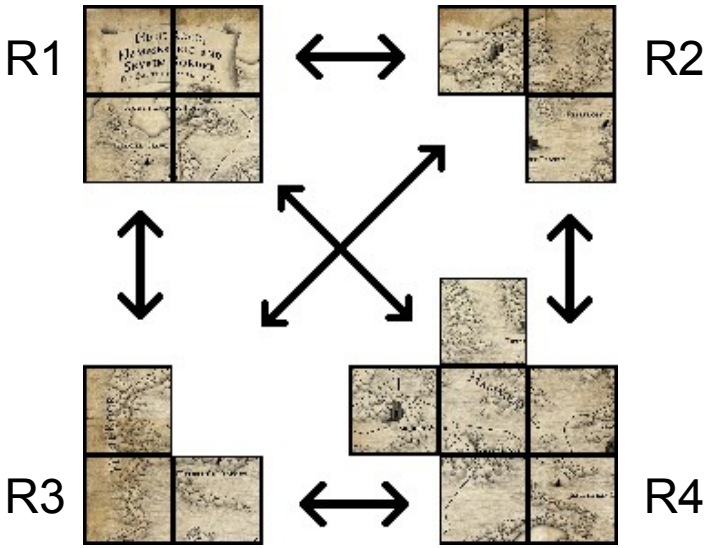
- Para efetuar o balanceamento de carga
  - Mundo do jogo é dividido em microcélulas – áreas de mesmo tamanho e com posição fixa no mundo do jogo



  - Representação em grafo:
    - Célula = vértice
    - Fronteira entre duas células = aresta
    - Região do mundo do jogo = partição do grafo
    - Peso do vértice = comunicação dentro da célula
    - Peso da aresta = comunicação entre as células que liga

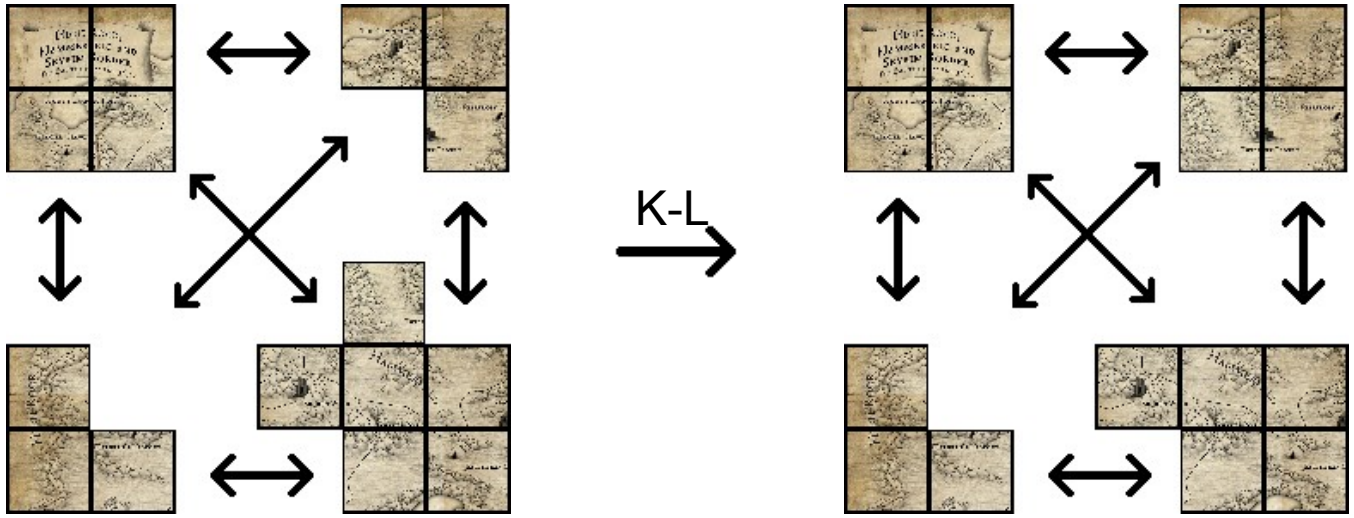
Criação das partições

- É criado um número de regiões do grafo igual ao de servidores disponíveis
  - Cada região é associada a um servidor
  - Começa-se associando a célula mais carregada, C_1, ao servidor com mais recursos, S_1
  - A próxima célula será a vizinha de C_1 ligada pela aresta de maior peso e assim por diante
  - Células vão sendo adicionadas à região até que a carga da região seja proporcional à capacidade do servidor
  - Passa-se para o próximo servidor com mais recursos, selecionando a célula livre mais carregada
- Cada região passa a ser administrada por um servidor
  - Servidores precisam trocar mensagens para que jogadores em diferentes regiões possam interagir



Refinamento das partições

- Com o passar do tempo, a distribuição de avatares e de hotspots pode mudar
  - É necessário refinar o particionamento
  - Utilização do algoritmo de Kernighan-Lin para reajustar o particionamento



Conclusão

Conclusão

- É possível, utilizando mais pesadamente o poder de processamento, reduzir-se o tráfego gerado por jogos MMOG, permitindo-se o barateamento da infra-estrutura cliente-servidor e aumentando o público do jogo.

Trabalhos futuros

Trabalhos Futuros

- Entrada e saída de nodos servidores
  - Particiona-se o ambiente virtual recursivamente à medida em que entram nodos
  - Quando um servidor sai, delega a sua região a um vizinho
- Topologia da rede *overlay* baseada na topologia real (Ratsanamy et al., 2002; Duong Ta, 2008)
- Difusão dos estados dos jogadores
  - Feita de maneira probabilística, baseado em difusão por push-gossip (Kermarrec et al., 2003)
    - Regiões mais próximas tenderão a ter o estado mais recente do jogador