

Princípios de comunicação de dados

Você já parou para observar a quantidade de jogos com componentes multijogador? E os aplicativos que conectam um mesmo jogo entre diversas plataformas? A indústria vem procurando jeitos de aprimorar, de formas diferentes, a experiência do usuário. Uma das grandes novidades no mercado são os jogos e aplicativos para dispositivos móveis que influenciam jogos de outras plataformas em tempo real, os *companions apps*!

Temos também os jogos de navegadores, onde as partidas são feitas nas páginas da internet, através de programas que nos permitem sites da web. Ainda acha pouco? Atualmente grandes empresas, como Sony e Microsoft, têm se esforçado para criar serviços que hospedam jogos na nuvem, para, posteriormente, serem alugados e jogados via *streaming*, através de consoles de última geração, como PS4 e Xbox One.

Para todas essas questões, a utilização de comunicação de dados é essencial, por isso iremos estudar e conhecer melhor esse assunto.



A imagem mostra a jogabilidade integrada de um jogo em duas plataformas, um tablet e um pc. No PC, o jogador controla um carro e o personagem, tem uma visualização traseira do carro, visualizando também ruas e obstáculos que estão à sua frente, a sensação do jogador é de estar conduzindo o veículo. O tablet faz o papel de um GPS, pois através dele o jogador visualiza o veículo, ruas e obstáculos de forma panorâmica. É uma forma de companion app do jogo Watch Dogs da Ubisoft, pois traz a integração do computador com o tablete.

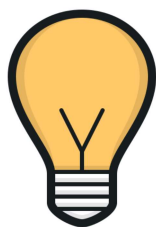
Figura 1: *Companion app* para o jogo *Watch Dogs* da *Ubisoft*. É possível jogar em uma plataforma e utilizar um dispositivo móvel para realizar ações no mundo virtual em tempo real.



É uma imagem do jogo do tiro em terceira pessoa *Mass Effect Infiltrator* para dispositivos móveis que é integrado com o mesmo jogo no computador. Desta forma a pontuação obtida no dispositivo móvel é refletida também no computador. Neste caso é uma integração assíncrona.

Figura 2: Jogo *Mass Effect: Infiltrator* para dispositivos móveis. Avanços no jogo influenciam quesitos de pontuação do jogo *Mass Effect 3* que afetam diretamente no final do jogo.

Você já deve ter percebido que a comunicação de dados, entre computadores e/ou plataformas, é um assunto complexo. É preciso avaliar cada jogo, no que se refere à comunicação, conforme a forma de conexão e os protocolos utilizados para troca de dados e, também, quanto à velocidade da rede. Além disso, temos os jogos para navegadores, que também demandam avaliação das plataformas e ferramentas que um jogo solicita de um navegador específico. Iremos abordar cada um dos assuntos individualmente, porém é importante perceber que todos estão interconectados.



Para saber mais sobre o *gameplay* do jogo *Watch Dogs* e seu *companion app*, acesse o *link* a seguir: <https://www.youtube.com/watch?v=gWDY4ge-xP8> (<https://www.youtube.com/watch?v=gWDY4ge-xP8>)

Antes de falarmos, mais detalhadamente, sobre cada um dos tópicos citados anteriormente, você precisa ter ideia de como funciona uma rede de computadores e precisa também conhecer os termos técnicos mais comuns relacionados a ela.



Para isso, imagine o seguinte exemplo: um jogo de tiro, jogado entre quatro usuários, através de um componente multijogador. Os quatro jogadores formam um time de caçadores em um mundo cheio de monstros perigosos, o objetivo do time é sobreviver aos inimigos até que o resgate chegue. Os monstros e as mudanças no clima do jogo são controlados por um computador central (servidor). Os caçadores são controlados pelos jogadores, cada um em seu computador, porém toda ação que um jogador realiza é comunicada diretamente ao servidor. O servidor, por sua vez, repassa as informações necessárias sobre o andamento da partida e seus personagens, para os outros jogadores.

Com esse exemplo em mente podemos entender alguns conceitos e alguns termos básicos:

- ◆ **Sistemas finais:** são os dispositivos que estão conectados à internet. No nosso exemplo são representados tanto pelos computadores que executam o jogo, quanto pelo computador central que controla os monstros. Também são conhecidos como hospedeiros (hosts), e, por vezes, subdivididos entre cliente e servidor.
- ◆ **Cliente:** computador que solicita um serviço através da rede; geralmente o cliente age a pedido de um ser humano, através de uma interface de usuário. No nosso exemplo de jogo, os computadores dos jogadores representam os clientes.
- ◆ **Servidor:** computador que realiza a parte principal do serviço, usando seus recursos locais e/ou outros serviços. Normalmente são máquinas mais poderosas. No nosso exemplo de jogo, o computador central representa um servidor.
- ◆ **Protocolo:** conjunto de regras que controla e possibilita conexão, comunicação e transferência de dados entre dois sistemas computacionais. Poderíamos dizer que o protocolo representa o idioma definido para a comunicação entre os sistemas finais.
- ◆ **Pacote:** conjunto de dados. No nosso exemplo de jogo são as informações trocadas entre clientes e servidor sobre os estados dos personagens, dos monstros e do mundo.
- ◆ **Aplicação de rede:** programas utilizam, em algum momento, a internet, ou uma rede local, para executar suas finalidades. No nosso exemplo, o componente multijogador representa uma aplicação de rede.
- ◆ **Largura de banda:** capacidade de transmissão de dados da conexão/rede; ou seja, a velocidade de envio e recebimento de dados. Poderíamos estimar, através de cálculos e testes, que a largura de banda recomendada para o nosso jogo seria em torno de 512kbps (512.000 bits por segundo).
- ◆ **Latência:** representa o atraso entre o envio e o recebimento de um pacote de dados entre dois sistemas finais, ou seja, o tempo necessário para que a informação percorra a rede e chegue até seu destino.

- ◆ **Roteamento:** refere-se ao processo de escolha de um caminho pelo qual serão enviados os pacotes de dados através da rede.

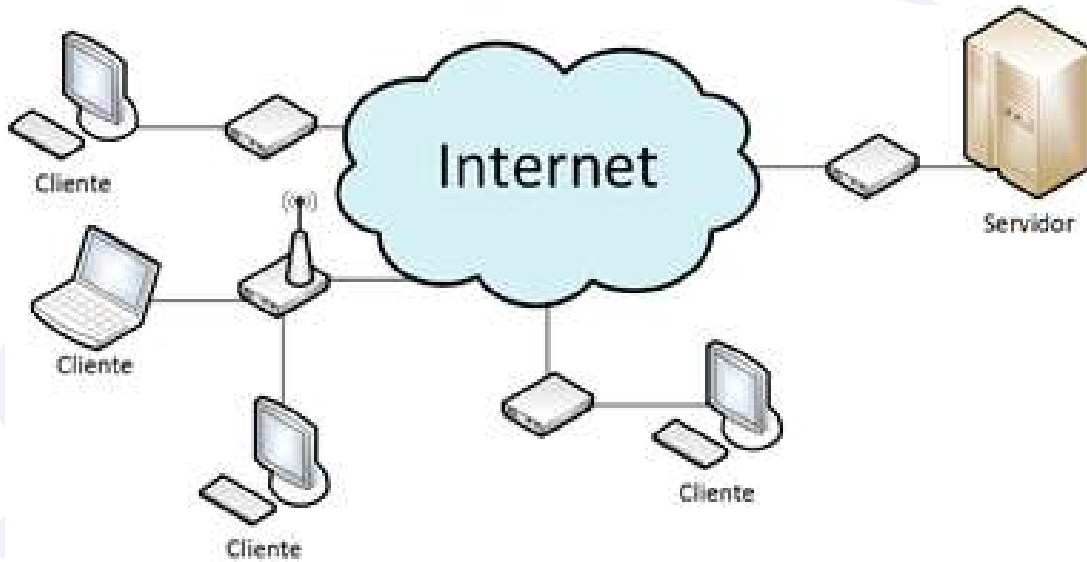


Figura 3: Esboço de uma rede para um jogo multijogador online.

Modelos de conexão

Quando se cria um jogo, e se faz necessário algum tipo de comunicação de dados, a primeira decisão que você deverá tomar é escolher o modelo de conexão de rede, também chamado de arquitetura da aplicação de rede. A arquitetura da aplicação é projetada pelo desenvolvedor e determina como a comunicação em uma aplicação de rede é organizada entre os vários sistemas finais. A escolha da arquitetura normalmente gira em torno de dois modelos de conexão: cliente-servidor e *peer-to-peer* (P2P).

Clique nas abas abaixo, para visualizar o conteúdo

1.1 Modelo cliente-servidor

O modelo cliente-servidor tem dois componentes: o servidor, responsável pela execução do serviço, e o cliente, responsável pela apresentação dos resultados e interação com o usuário. Nesta arquitetura sempre existe um servidor em funcionamento esperando para atender às requisições dos clientes. Observe que, na arquitetura cliente-servidor, os clientes não se comunicam diretamente uns com os outros. Você já deve ter percebido que o nosso exemplo de jogo utiliza esse modelo de conexão, pois os computadores dos jogadores são os clientes e o computador central é o servidor do jogo.



Figura 4: Conexões dentro do modelo cliente/servidor

1.2 Modelo *peer-to-peer* (P2P)

No modelo *peer-to-peer* (P2P) puro não existe um sistema central - um servidor - intermediando a comunicação entre os diversos jogadores (clientes). Nesse modelo, todo jogador estabelece, diretamente, conexão com todos os outros jogadores. Nesta arquitetura todos os participantes são ao mesmo tempo servidores (oferecem serviços e recursos) e clientes (usam serviços e recursos) uns dos outros. Esse modelo experimenta um atraso pequeno no envio de dados (latência) entre os sistemas finais, devido à conexão direta entre um jogador e outro. Olhando a figura podemos ver que o número total de conexões entre os jogadores aumenta de forma rápida, conforme mais usuários são inseridos na aplicação (figura 5), trazendo, desta forma, um custo mínimo de requisitos da largura de banda (capacidade de transmissão de dados da conexão/rede). Outro ponto diz respeito à complexidade de manutenção das conexões, devido a sua natureza altamente distribuída e descentralizada pode ser difícil gerenciar aplicações P2P.

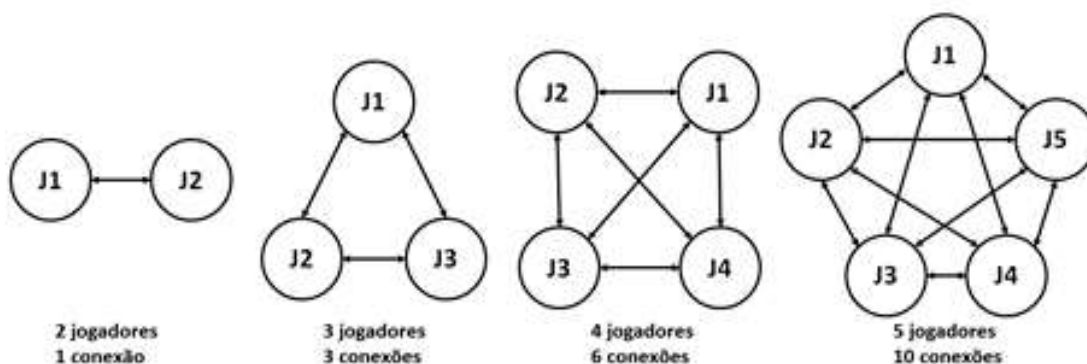


Figura 5: Conexões dentro do modelo peer-to-peer (P2P).



Clique no link a seguir para saber mais sobre como construir um jogo com conexão P2P (inglês)

<http://gamedevelopment.tutsplus.com/tutorials/building-a-peer-to-peer-multiplayer-networked-game--gamedev-10074>

(<http://gamedevelopment.tutsplus.com/tutorials/building-a-peer-to-peer-multiplayer-networked-game--gamedev-10074>)

Poderíamos usar como exemplo um jogo de luta. Neste tipo de jogo não existem monstros, todos os personagens são controlados pelos jogadores da partida, dessa forma, não é necessário um servidor para controlar as ações de inimigos ou do estado do mundo virtual. Assim, tornamos a comunicação mais veloz, tendo em vista que a conexão direta entre clientes gera uma latência menor.



Agora, você já viu que para dois jogadores se comunicarem é necessário que, de alguma forma, eles possam enviar informações um para o outro, seja de forma direta (P2P) ou por intermédio de um servidor. Porém, somente isso não é o bastante, para que a comunicação seja realizada é necessário que todos utilizem o mesmo protocolo, ou seja, os dados trocados precisam atender um conjunto de regras.

Protocolos

Podemos pensar em rede de computadores como: diversas máquinas interligadas fisicamente entre si, através das quais utilizadores promovem a troca de informação que julgam relevantes. O Protocolo é um formato definido para transferência de dados entre esses dispositivos. É difícil generalizar sobre protocolos, pois eles variam muito em propósito e sofisticação. A maioria dos protocolos especifica uma ou mais das seguintes propriedades:

- ◆ Endereçamento
- ◆ Tamanho e forma do pacote de dados
- ◆ Formas de reconhecimento
- ◆ Verificação e correção de erros
- ◆ Encriptação de dados
- ◆ Compressão de dados
- ◆ Roteamento
- ◆ Mapeamento de endereço

Você já deve ter ouvido o nome de alguns dos principais protocolos utilizados na comunicação entre dispositivos:

Clique nas abas abaixo, para visualizar o conteúdo

IP (*Internet Protocol*)

Usado entre todas as máquinas em rede para encaminhamento dos dados. O propósito fundamental do protocolo IP é o roteamento de pacotes através de um conjunto de redes interconectadas. Imagine que em um jogo as informações diversas devem ser enviadas para os jogadores corretos, é através do protocolo IP que se torna possível a identificação dos jogadores na rede.

HTTP (*Hyper Text Transfer Protocol*)

É o protocolo utilizado para controlar a comunicação entre o servidor de Internet e o navegador. Este protocolo deve ser usado caso o jogo seja executado em um navegador.

TCP (*Transfer Control Protocol*)

O TCP permite dar segurança à transferência de informações e verificar se elas foram bem recebidas pelo destinatário. Caso contrário, volta a enviar os dados. Esse protocolo garantiria que a comunicação entre jogadores aconteceria sem erros ou perda

de dados.

UDP (*User Datagram Protocol*)

O UDP é um protocolo de transporte de informação, mas não é tão confiável como TCP. Não há qualquer tipo de garantia que o pacote de dados irá chegar, o que o torna mais rápido. Esse protocolo tornaria a comunicação entre jogadores mais rápida, porém não garantiria que as informações fossem repassadas em ordem ou até mesmo que chegassem a seus destinos.



Para saber mais sobre Protocolos UDP e TCP para jogos online (Inglês), acesse:

<http://gafferongames.com/networking-for-game-programmers/udp-vs-tcp/>

(<http://gafferongames.com/networking-for-game-programmers/udp-vs-tcp/>)

Navegadores

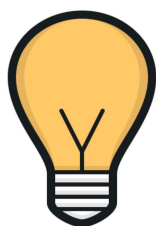
Tem sido cada vez mais comum encontrar jogos que rodam direto do navegador (também conhecidos como *browser*). O que teve início com pequenos desafios em Flash, ganhou proporções inimagináveis e hoje não é difícil encontrar jogos grandes, bem feitos e de alta qualidade, que funcionam a partir do navegador. As principais tecnologias utilizadas para o desenvolvimento de jogos para navegador são: SVG (*Scalable Vector Graphics*), Canvas (um elemento do HTML5), WebGL, Adobe Flash, Unity Web Player e Java. Atualmente, os navegadores mais novos, como Chrome, Opera e Firefox, são compatíveis com tais tecnologias.



Figura 6: O jogo InveciCar Online utiliza o plug-in Unity Web Player para rodar em navegadores.

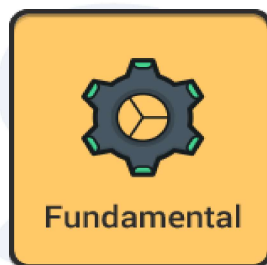
Um dos pontos positivos dos jogos executados em navegadores é que todas as informações do jogo são recebidas pelo navegador através da internet, dessa forma não precisamos instalar nenhum programa extra para participar de uma partida. A troca de dados é feita através do protocolo HTTP.

O HTTP segue também o modelo cliente-servidor. O cliente, normalmente o navegador, estabelece uma conexão com um servidor HTTP. Ao estabelecer uma conexão, o cliente procede com a requisição de recursos, estes que são transferidos do servidor para o cliente. Após receber os recursos solicitados, o navegador inicia o processo de apresentação do conteúdo. Caso seja necessário, o cliente solicita novamente outros recursos ao servidor HTTP.



Para saber mais sobre jogos de navegadores, clique no link a seguir e leia o artigo:

<http://www.tecmundo.com.br/web/1978-tendencia-jogos-de-alta-qualidade-via-navegador.htm> (<http://www.tecmundo.com.br/web/1978-tendencia-jogos-de-alta-qualidade-via-navegador.htm>)



Você pode conhecer, neste material, as modalidades de jogos multijogador, jogos com os companion apps e jogos web. Eles demandam tecnologias como os protocolos e as arquiteturas de redes para serem jogados e tecnologias para rodar nos navegadores web. Com isso, você pode avaliar a necessidade técnica envolvida no projeto de jogo, de acordo com estas características. A habilidade necessária para programar, para determinar quais tecnologias e a forma de aplicá-las no desenvolvimento do jogo, será trabalhada mais adiante no curso; tendo como foco os jogos multijogador.

Referências bibliográficas

Carissimi, Alexandre S; Rochol, Juergen e Graville, Lisandro. **Redes de Computadores**. Editora: Bookman, 2009.

Kurose, James F. e Ross, Keith W. Ross. **Redes de Computadores e a Internet**. 3ª edição. Editora: Pearson, 2005.

Rabin, Steve. **Introdução ao Desenvolvimento de Games**. Vol.2, 2ª edição. Editora: Cengage, 2013.



Senac