Indice

1. Esquema Cliente-Servidor
   1. Cliente
      1. Requisição de conexão
      2. Requisição da lista de clientes
      3. Requisição de desconexão
   2. Servidor
      1. Estabelecendo a conexão
      2. Fornecendo lista de clientes
      3. Estabelecendo desconexão
2. Usando o Chat
   1. Comunicação TCP
   2. Threads
   3. Parte Remetente do Chat
   4. Parte Destinatário do Chat
3. Transferindo Arquivo
   1. Características da Aplicação
   2. Estrutura do pacote
   3. Estrutura do ACK
   4. Parte Remetente
      1. Segmentação do Arquivo
      2. Janela de Remetente
      3. Controle de Fluxo do Remetente
      4. Confiabilidade no Remetente
   5. Parte Destinatário
      1. Remontagem do Arquivo
      2. Janela de Destinatario
      3. Controle de Fluxo do Destinatário
      4. Confiabilidade no Destinatário
   6. Controle de Congestionamento
4. Arquitetura P2P
5. Arquitetura Cliente-Servidor
6. Package de Aplicação
7. Package de Confiabilidade
8. Package de Controle
9. Package de Transporte
10. Package de TransporteTCP
11. Package de TransporteUDP

Esquema Cliente-Servidor

# Cliente

## Requisição de conexão

O cliente, através da interface gráfica, faz uma requisição de conexão para o servidor enviando uma mensagem TCP com o tipo de requisição para conexão e o login e senha em seguida.

## Requisição da lista de clientes

Para utilizar o chat e enviar arquivos, o usuário precisa saber quais usuários estão on-line. Para obter uma lista de usuários on-line ele manda uma mensagem de controle TCP para o servidor requisitando tal lista. Ele precisa manter essa lista sempre atualizada para tanto o uso do chat quanto o envio de arquivos para os outros usuários.

## Requisição de desconexão

Da mesma maneira que a conexão, o cliente manda uma mensagem TCP com tipo de requisição para desconexão em seguida o login.

# Servidor

## Estabelecendo a conexão

Ao receber uma mensagem TCP do tipo conexão de algum cliente, o servidor irá checar em seu banco de dados (arquivo), se o login e senha estão corretos para em seguida confirmar o status de conectado ao cliente. Caso o login ou senha estejam incorretos, o servidor enviará uma mensagem TCP sinalizando o erro.

## Fornecendo lista de clientes

Depois que o usuário conecta-se ao servidor, o servidor receberá mensagens TCP do tipo status, que será uma requisição da lista de clientes on-line que ele possui no sistema.

## Estabelecendo desconexão

Ao receber uma mensagem TCP do tipo desconexão, o servidor buscará o login do usuário que requisitou a desconexão e mudará seu status na lista de clientes para off-line e fechará a porta de conexão.

Usando o Chat

# Comunicação TCP

Toda comunicação via chat é feita através de TCP. Como vários usuários podem estar on-line ao mesmo tempo, é necessária a escrita em várias portas para envio a esses usuários. Então um laço é realizado para que a mesma mensagem seja enviada para todos os usuários da lista de clientes obtidas previamente do servidor.

# Threads

Como podem existir vários clientes conectados, e todos eles podem estar enviado arquivos, é necessário que um cliente esteja apto a receber vários arquivos ao mesmo tempo. Isso é realizado através de threads, ou seja, para cada cliente da lista de cliente que tentar enviar um arquivo, será criada uma thread e, em cada uma delas, o recebimento do arquivo é habilitado.

Transferindo Arquivo

# Características da Aplicação

A aplicação possui um buffer de recepção para cada arquivo enviado, possuindo tamanho máximo de 64 Kbytes. O cliente só poderá enviar 1 arquivo por vez, repassando-o para todos os usuários e poderá receber *n* arquivos dos *n* usuários da aplicação.

# Estrutura do pacote

O pacote será implementado como segue na figura abaixo. A estrutura do pacote contém os seguintes campos:

* Campo Flag (1 byte): Este campo indica se ainda existem pacotes a serem enviados, se o valor da Flag for 1, significa que existem mais pacotes para serem enviados, caso contrário, não existem mais pacotes para enviar, o pacote que possui Flag 0 é o último pacote do arquivo;
* Campo Tamanho (2 bytes): Este campo representa o tamanho total do pacote;
* Campo numSeq (4 bytes): Este campo contém o número de seqüência do pacote.
* Campo de dados: Este campo possui os dados úteis à aplicação.



O cabeçalho terá tamanho fixo de 7 bytes e o tamanho máximo do campo de dados será de 1000 bytes. Logo, o tamanho máximo do pacote será de 1007 bytes.

# Estrutura do ACK

O ACK será mandado via TCP. Portanto, será enviada uma mensagem contendo o valor do ACK e o tamanho da janela de recepção.

# Parte do remetente

## Segmentação do Arquivo

Para envio do arquivo, primeiramente é necessário a sua divisão em blocos. Isso se deve ao fato do arquivo poder ter um tamanho muito grande e estourar o heap (parte da memória reservada para alocação dinâmica do programa). Depois de pegar apenas um desses blocos, que garantidamente não haverá estouro de heap, essa parcela será ainda quebrada em vários arrays de bytes, e cada um desses arrays será um pacote a ser enviado.

## Janela de Remetente

A Janela de Remetente é implementada para o serviço de confiabilidade na transferência do arquivo utilizando protocolo de comunicação UDP. Nela está registrado o status de cada pacote a ser enviado pertencente a essa janela. No caso da Janela de Remetente temos os possíveis status:

1. Pacote já reconhecido;
2. Pacote enviado e ainda não reconhecido;
3. Pacote autorizado e ainda não enviado;
4. Pacote não autorizado;

A confiabilidade da parte do Remetente atualiza a Janela de Remetente sempre que é enviado algum pacote, ou então um ACK de algum dos pacotes enviados é recebido, alterando o status da posição da janela correspondente ao pacote de acordo com cada evento descrito. Para a operação de atualização, são feitas operações para verificar se o pacote pertence à Janela de Remetente e para saber se o ACK recebido é para o pacote que corresponde à posição base da Janela.

## Controle de Fluxo do Remetente

Para não provocar um transbordamento no buffer da aplicação do destinatário pelo excessivo envio de pacotes, foi implementado na aplicação o serviço de Controle de Fluxo, nesse caso será explicado a funcionalidade do lado remetente do Controle de Fluxo da aplicação.

Esse lado do Controle registra informações sobre o último byte enviado, o último byte reconhecido (último ACK recebido) e sobre o tamanho da janela de recepção do destinatário, informação enviada pela aplicação destinatária durante a confirmação do recebimento de cada pacote enviado pelo remetente, toda vez que um novo pacote for enviado ou quando um ACK for recebido. Através desses registros será criada uma restrição que, juntamente com a restrição do Controle de Congestionamento, irá manipular a quantidade de pacotes enviados pela aplicação remetente. A restrição do Controle de Fluxo do Remetente é a mesma utilizada pelo Controle de Fluxo do Remetente do protocolo TCP.

# Parte do remetente

## Remontador do Arquivo

Para o recebimento do arquivo, primeiramente é necessário transformar um pacote recebido em array de bytes. Depois disso será montado um bloco de arrays de bytes, esse bloco corresponderá a uma parte do arquivo original. Juntando cada um dos blocos em um arquivo final, teremos, no lado destinatário, o arquivo original enviado.

## Janela de Destinatário

A Janela de Destinatário é implementada para o serviço de confiabilidade na transferência do arquivo utilizando protocolo de comunicação UDP e no Controle de Fluxo do Destinatário. Nela está registrado o status de cada pacote a ser recebido pertencente à essa janela. No caso da Janela de Destinatário temos os possíveis status:

1. Pacote fora da ordem já reconhecido;
2. Pacote aguardado ainda não recebido;
3. Pacote aceitável;
4. Pacote não autorizado;

A confiabilidade da parte do Destinatário utiliza a Janela de Destinatário sempre que é recebido algum pacote, nesse caso a janela é atualizada alterando o status da posição da janela correspondente ao pacote recebido, ou então para enviar o ACK pertencente ao pacote com o número de seqüência esperado. Para a operação de atualização, são feitas operações para verificar se o pacote pertence à Janela de Destinatário e para a operação de enviar ACK, será calculado através da Janela de Destinatário, o número de seqüência esperado pela aplicação do próximo pacote.

O Controle de Fluxo do Destinatário utiliza a Janela de Destinatário para saber quais pacotes ainda não foram recebidos, bastando apenas consultar os status das posições correspondentes aos pacotes aguardados e ainda não recebidos. Esse interesse pela Janela de Destinatário será explicado no tópico seguinte.

## Controle de Fluxo do Destinatário

Agora será explicado a funcionalidade do lado destinatário do Controle de Fluxo da aplicação, para concluir a explicação da implementação do serviço de Controle de Fluxo.

Esse lado do Controle irá registrar informações sobre o último byte recebido pelo buffer (podendo pertencer a um pacote fora de ordem), o último byte lido do buffer e sobre o tamanho da janela de recepção, informação calculada e enviada para a aplicação remetente durante o envio dos ACKs, toda vez que um novo pacote for adicionado no buffer ou quando a aplicação fizer uma requisição de leitura do buffer, o qual removerá os dados logo após serem lidos. Através desses registros e da Janela de Destinatário será calculado o tamanho da janela de recepção. Considerando que os pacotes chegam fora de ordem , não podemos contar apenas com o último byte recebido, pois caso ele pertença a um pacote fora de ordem, iremos calcular a janela considerando que os pacotes antes dele já foram recebidos, o que não necessariamente é verdade. Para resolver este impasse, consulta-se a Janela de Destinatário, com ela saberemos quantos pacotes ainda não chegaram desde que o último byte recebido fora de ordem foi adicionado no buffer. Depois disso, é calculado o tamanho desses pacotes e com esse tamanho, juntamente com o valor do último byte recebido e o último byte lido, iremos efetivamente calcular o tamanho da janela de recepção.

# Controle de Congestionamento

Com o intuito de não provocar uma sobrecarga na rede pelo envio excessivo de pacotes, foi implementado na aplicação o serviço de Controle de Congestionamento, que monitora os intervalos de envio de cada pacote e de chegada dos seus respectivos ACKs.

Esses monitoramentos irão servir para calcular as amostragens dos RTTs, por conseguinte, estes servirão para calcular os valores Estimados de RTT, que servirão para calcular os valores de DevRTT, e por fim estabelecer os melhores valores de Timeout dos pacotes.

Ao serem identificadas perdas de pacotes (caso onde são recebidos três ACKs repetidos), a restrição sobre a quantidade de pacotes a serem enviados pela aplicação (janela de envio) é automaticamente mudada pelo serviço de Controle de Congestionamento, utilizando o algoritmo de controle de congestionamento TCP Reno.

Arquitetura P2P



Arquitetura Cliente-Servidor



Package de Aplicação



Package de Confiabilidade



Package de Controle



Package de Transporte



Package de TransporteTCP



Package de TransporteUDP

