

Atividade 2 - Programação com Sockets UDP

Sistemas Distribuídos – Prof. Rodrigo Campiolo

Questão 1) Fazer um chat P2P que possibilite os clientes trocarem mensagens entre si.

As mensagens possuem o formato:

- tamanho apelido (tam_apl) [1 byte]
- apelido [tam_apl bytes]
- tamanho mensagem (tam_msg) [1 byte]
- mensagem [tam_msg bytes]

```
      1 byte                      0 - 255 bytes
+++++-----+-----+-----+-----+-----+-----+-----+-----+
| Nick Size |                               Nick [Nick Size] |
+++++-----+-----+-----+-----+-----+-----+-----+-----+
| Message Size |                         Message [Message Size] |
+++++-----+-----+-----+-----+-----+-----+-----+-----+
```

Opcionais:

- fazer um servidor TCP para acessar e incluir o apelido e endereço da lista de pessoas ativas na conversação.
- ao fechar a conexão com o servidor TCP, deve-se remover o apelido e endereço da lista.
- a cada 2s os clientes podem verificar se há modificações na lista do servidor ou, o servidor pode enviar uma notificação de atualização sempre que ocorrer uma mudança de estado.
- fazer um mecanismo de *emotions* usando ícones .

Questão 2) Fazer um sistema de upload de arquivos via UDP. Um servidor UDP deverá receber as partes dos arquivos (1024 bytes), verificar ao final a integridade via um checksum (MD5) e armazenar o arquivo em uma pasta padrão. *Sugestões:* o servidor pode receber o nome e tamanho do arquivo como o primeiro pacote e o checksum como o último. Testar o servidor com arquivos textos e binários (ex: imagens, pdf) de tamanhos arbitrários (ex: 100 bytes, 4KiB, 4MiB). O protocolo para a comunicação deve ser criado e especificado textualmente ou graficamente.

Opcionais:

- gerar log no servidor com o horário de início e término da transferência, nome do arquivo, tamanho do arquivo, origem do arquivo.
- fazer interface gráfica no cliente para selecionar localmente o arquivo.
- fazer barra de estado para mostrar o percentual transferido e o restante.
- suportar transferências simultâneas.