

# Relatório do Trabalho 2 de Redes

Peterson Carvalho (GRR20163053) e Luiz Chiconatto (GRR20166814)

13 de Outubro de 2017

## 1 Introdução

Esse trabalho tem como objetivo implementar um jogo de batalha naval com 4 jogadores conectados em uma rede em anel utilizando a linguagem C com socket datagram - UDP.

## 2 Corpo da Mensagem

As mensagens tem o padrão conforme a Tabela 1 e possuem os campos: Delimitador de Início, Controle de Acesso, Controle de Mensagem, Endereço de Destino, Endereço de Origem, Dados, CRC, Delimitador de Fim e Status.

Para o bastão, a mensagem tem o formato conforme a Tabela 2.

Table 1: Corpo da Mensagem

Delimitador de Início	Controle de Acesso	Controle de Mensagem	Endereço de Destino	Endereço de Origem	Dados	CRC	Delimitador de Fim	Status
8 bits	8 bits	8 bits	48 bits	48 bits	18200 bytes	32 bits	8 bits	8 bits

Table 2: Corpo do Bastão

Delimitador de Início	Controle de Acesso	Delimitador de Fim
8 bits	8 bits	8 bits

## 3 Formato dos Dados

Os dados seguem o formato da Tabela 3. O primeiro campo é o campo de tipo de mensagem, já que a mensagem pode ser de ataque, de alerta de navio afundado que é mostrado para todos os outros players ou a mensagem inicial de posicionamento dos navios do início do jogo. Os dois campos seguintes são as coordenadas x e y do ataque (esses campos ficam vazios caso a mensagem não seja do tipo Ataque). O quarto campo é o campo que diz qual o player que a

mensagem de alerta está se referindo e o quinto campo é o campo se diz se esse mesmo player morreu ou não. Os 18192 bytes seguintes não são usados.

Table 3: Formato dos Dados

Tipo da Mensagem	Coordenada X do ataque	Coordenada Y do ataque	Número do jogador referente ao alerta	Jogador morreu?	Dados sem uso
8 bits	8 bits	8 bits	8 bits	8 bits	18192 bytes

## 4 Status

O campo de Status segue o formato da Tabela 4. Todo esse campo de status é mudado apenas pelo destinatário da mensagem quando ele a recebe. O primeiro bit é acionado quando a mensagem chega e o segundo bit é acionado quando a mensagem chega sem erros. O terceiro bit é mudado para 1 pelo destinatário tem o navio acertado e o quarto bit é acionado quando o navio inteiro acaba de afundar. O quinto campo tem 3 bits e diz o número do jogador que teve o navio afundado. O último bit diz se o player acaba de morrer ou não.

Table 4: Formato do Status

Mensagem chegou	ACK/NACK	Resultado	Navio afundou?	Player que teve o navio afundado	Player morreu?
1 bit	1 bit	1 bit	1 bit	3 bits	1 bit

## 5 Funcionamento do Programa

Cada player tem seu proprio codigo, que são basicamente todos iguais só mudando os IPs pra cada máquina. O player 1 deve inicializar o jogo mandando uma mensagem do tipo Fase de Posicionamento, que indica a hora de cada player posicionar os seus navios. Depois disso, o primeiro jogador faz o primeiro ataque a algum player e envia para o anel. A partir daí, o jogo segue o *loop*. Quando o jogador recebe uma mensagem, ele tem que verificar se aquela mensagem é para ele ou veio dele pelos endereços contidos nos campos da mensagem. Caso não seja, ele apenas passa adianta a mensagem.

Se o player recebe uma mensagem destinada a ele, verifica-se o tipo da mensagem e faz as ações apropriadas.

Se o player recebe uma mensagem que veio dele, então ele verifica o status de ack e chegada da mensagem e então os resultados (acertou, matou, etc) caso esta mensagem for de ataque.

## **6 Timeout**

Caso a mensagem demore mais que 2 segundos para chegar de volta a quem enviou, a mensagem é enviada novamente até obter sucesso.