

ddt - RCI

Generated by Doxygen 1.8.9.1

Wed Mar 18 2015 15:10:57



# Contents

<b>1</b>	<b>Data Structure Index</b>	<b>1</b>
1.1	Data Structures . . . . .	1
<b>2</b>	<b>File Index</b>	<b>3</b>
2.1	File List . . . . .	3
<b>3</b>	<b>Data Structure Documentation</b>	<b>5</b>
3.1	args_parse_options Struct Reference . . . . .	5
3.1.1	Field Documentation . . . . .	5
3.1.1.1	argc . . . . .	5
3.1.1.2	argv . . . . .	5
3.1.1.3	startup_data . . . . .	5
3.2	peer_data Struct Reference . . . . .	5
3.2.1	Field Documentation . . . . .	6
3.2.1.1	id . . . . .	6
3.2.1.2	node . . . . .	6
3.2.1.3	service . . . . .	6
3.2.1.4	socket . . . . .	6
3.3	startup_data Struct Reference . . . . .	6
3.3.1	Field Documentation . . . . .	6
3.3.1.1	dest_size . . . . .	6
3.3.1.2	destination . . . . .	6
3.3.1.3	family . . . . .	6
3.3.1.4	ringport . . . . .	6
3.4	transversal_data Struct Reference . . . . .	7
3.4.1	Field Documentation . . . . .	7
3.4.1.1	ext_addr . . . . .	7
3.4.1.2	id . . . . .	7
3.4.1.3	peer_pred . . . . .	7
3.4.1.4	peer_succ . . . . .	7
3.4.1.5	reg . . . . .	7
3.4.1.6	ring . . . . .	7

3.4.1.7	<a href="#">serv_arrang</a>	7
3.4.1.8	<a href="#">socket_with_new_node</a>	7
3.4.1.9	<a href="#">startup_data</a>	7
3.4.1.10	<a href="#">t</a>	7
3.4.1.11	<a href="#">u</a>	7
<b>4</b>	<b>File Documentation</b>	<b>9</b>
4.1	<a href="#">include/common.h File Reference</a>	9
4.2	<a href="#">include/interface.h File Reference</a>	9
4.2.1	<a href="#">Function Documentation</a>	9
4.2.1.1	<a href="#">interface</a>	9
4.2.1.2	<a href="#">print_ui</a>	10
4.3	<a href="#">include/net_common.h File Reference</a>	10
4.3.1	<a href="#">Function Documentation</a>	10
4.3.1.1	<a href="#">getsockaddr</a>	10
4.4	<a href="#">include/net_tcp.h File Reference</a>	10
4.4.1	<a href="#">Function Documentation</a>	10
4.4.1.1	<a href="#">connect_tcp</a>	10
4.4.1.2	<a href="#">createserver_tcp</a>	11
4.5	<a href="#">include/net_udp.h File Reference</a>	11
4.5.1	<a href="#">Function Documentation</a>	11
4.5.1.1	<a href="#">createsocket_udp</a>	11
4.5.1.2	<a href="#">getudpdest</a>	11
4.6	<a href="#">include/options.h File Reference</a>	11
4.6.1	<a href="#">Function Documentation</a>	12
4.6.1.1	<a href="#">parse_options</a>	12
4.7	<a href="#">include/ring.h File Reference</a>	12
4.7.1	<a href="#">Function Documentation</a>	12
4.7.1.1	<a href="#">join_ring</a>	12
4.8	<a href="#">include/trata_message.h File Reference</a>	12
4.8.1	<a href="#">Function Documentation</a>	12
4.8.1.1	<a href="#">check_message</a>	12
4.8.1.2	<a href="#">dist</a>	12
4.8.1.3	<a href="#">preenche_predi_info</a>	12
4.8.1.4	<a href="#">trata_mensagem</a>	12
4.8.1.5	<a href="#">verifica_se_responsavel</a>	12
4.8.1.6	<a href="#">write_message_tcp</a>	12
4.9	<a href="#">src/interface.c File Reference</a>	12
4.9.1	<a href="#">Detailed Description</a>	13
4.9.2	<a href="#">Macro Definition Documentation</a>	13

4.9.2.1	<a href="#">_XOPEN_SOURCE</a>	13
4.9.3	<a href="#">Function Documentation</a>	13
4.9.3.1	<a href="#">interface</a>	13
4.9.3.2	<a href="#">print_error</a>	13
4.9.3.3	<a href="#">print_join_l</a>	14
4.9.3.4	<a href="#">print_join_s</a>	14
4.9.3.5	<a href="#">print_search</a>	14
4.9.3.6	<a href="#">print_ui</a>	14
4.10	<a href="#">src/main.c File Reference</a>	14
4.10.1	<a href="#">Detailed Description</a>	14
4.10.2	<a href="#">Function Documentation</a>	14
4.10.2.1	<a href="#">main</a>	14
4.11	<a href="#">src/net_common.c File Reference</a>	14
4.11.1	<a href="#">Macro Definition Documentation</a>	15
4.11.1.1	<a href="#">_XOPEN_SOURCE</a>	15
4.11.2	<a href="#">Function Documentation</a>	15
4.11.2.1	<a href="#">getsockaddr</a>	15
4.12	<a href="#">src/net_tcp.c File Reference</a>	15
4.12.1	<a href="#">Macro Definition Documentation</a>	16
4.12.1.1	<a href="#">RCI_BACKLOG</a>	16
4.12.2	<a href="#">Function Documentation</a>	16
4.12.2.1	<a href="#">connect_tcp</a>	16
4.12.2.2	<a href="#">createserver_tcp</a>	16
4.13	<a href="#">src/net_udp.c File Reference</a>	16
4.13.1	<a href="#">Function Documentation</a>	16
4.13.1.1	<a href="#">createsocket_udp</a>	16
4.13.1.2	<a href="#">getudpdest</a>	17
4.14	<a href="#">src/options.c File Reference</a>	17
4.14.1	<a href="#">Detailed Description</a>	17
4.14.2	<a href="#">Macro Definition Documentation</a>	17
4.14.2.1	<a href="#">_XOPEN_SOURCE</a>	17
4.14.3	<a href="#">Function Documentation</a>	17
4.14.3.1	<a href="#">parse_options</a>	17
4.14.3.2	<a href="#">usage</a>	17
4.15	<a href="#">src/ring.c File Reference</a>	18
4.15.1	<a href="#">Macro Definition Documentation</a>	18
4.15.1.1	<a href="#">RCI_MSGSIZE</a>	18
4.15.2	<a href="#">Function Documentation</a>	18
4.15.2.1	<a href="#">join_ring</a>	18
4.16	<a href="#">src/trata_message.c File Reference</a>	18

---

4.16.1	Function Documentation . . . . .	18
4.16.1.1	check_message . . . . .	18
4.16.1.2	dist . . . . .	18
4.16.1.3	preenche_predi_info . . . . .	19
4.16.1.4	trata_mensagem . . . . .	19
4.16.1.5	verifica_se_responsavel . . . . .	19
4.16.1.6	write_message_tcp . . . . .	19
 <b>Index</b>		 <b>21</b>

# Chapter 1

## Data Structure Index

### 1.1 Data Structures

Here are the data structures with brief descriptions:

<a href="#">args_parse_options</a>	Estrutura com os argumentos para a função <a href="#">parse_options</a> . . . . .	5
<a href="#">peer_data</a>	Estrutura com a informação referente a um par . . . . .	5
<a href="#">startup_data</a>	Estrutura com os dados a ser recolhidos pela linha de comandos . . . . .	6
<a href="#">transversal_data</a>	Estrutura com as referências necessárias aos vários módulos do projecto . . . . .	7





## Chapter 2

# File Index

### 2.1 File List

Here is a list of all files with brief descriptions:

include/common.h	
Cabeçalho com declarações das estruturas de dados transversais aos vários módulos do projecto . . . . .	9
include/interface.h	
Cabeçalhos e protótipos associados às funções relacionadas com a interface com o utilizador . . . . .	9
include/net_common.h	
Cabeçalhos e protótipos associados às funções relacionadas com a rede mas não associadas a nenhum protocolo em específico . . . . .	10
include/net_tcp.h	
Cabeçalhos e protótipos associados às funções relacionadas com o protocolo TCP . . . . .	10
include/net_udp.h	
Cabeçalhos e protótipos associados às funções relacionadas com o protocolo UDP . . . . .	11
include/options.h	
Cabeçalhos e protótipos associados ao módulo de processamento das opções de linha de comandos . . . . .	11
include/ring.h	
. . . . .	12
include/trata_message.h	
. . . . .	12
src/interface.c	
Interface com o utilizador . . . . .	12
src/main.c	
Ficheiro principal do projecto . . . . .	14
src/net_common.c	
Ficheiro responsável pela implementação das funções relacionadas com rede mas não associadas a nenhum dos protocolos utilizados . . . . .	14
src/net_tcp.c	
Ficheiro responsável pela implementação das funções de rede associadas ao protocolo TCP . . . . .	15
src/net_udp.c	
Ficheiro responsável pela implementação das funções de rede associadas ao protocolo UDP . . . . .	16
src/options.c	
Ficheiro responsável pela interpretação dos argumentos passados via linha de comandos . . . . .	17
src/ring.c	
. . . . .	18
src/trata_message.c	
. . . . .	18



## Chapter 3

# Data Structure Documentation

### 3.1 args\_parse\_options Struct Reference

Estrutura com os argumentos para a função [parse\\_options](#).

```
#include <options.h>
```

#### Data Fields

- int \* [argc](#)  
*Ponteiro para o local em memória do contador de argumentos.*
- char \*\*\* [argv](#)  
*Ponteiro para o local em memória dos argumentos.*
- struct [startup\\_data](#) \* [startup\\_data](#)  
*Ponteiro para uma das estruturas transversais ao projecto.*

#### 3.1.1 Field Documentation

3.1.1.1 int\* args\_parse\_options::argc

3.1.1.2 char\*\*\* args\_parse\_options::argv

3.1.1.3 struct startup\_data\* args\_parse\_options::startup\_data

The documentation for this struct was generated from the following file:

- include/[options.h](#)

### 3.2 peer\_data Struct Reference

Estrutura com a informação referente a um par.

```
#include <common.h>
```

#### Data Fields

- int [id](#)  
*Número do nó pelo qual o par é responsável.*

- char [node](#) [256]  
*Endereço do par.*
- char [service](#) [16]  
*Porto do par.*
- int [socket](#)  
*Socket TCP com o par.*

### 3.2.1 Field Documentation

3.2.1.1 int peer\_data::id

3.2.1.2 char peer\_data::node[256]

3.2.1.3 char peer\_data::service[16]

3.2.1.4 int peer\_data::socket

The documentation for this struct was generated from the following file:

- include/[common.h](#)

## 3.3 startup\_data Struct Reference

Estrutura com os dados a ser recolhidos pela linha de comandos.

```
#include <common.h>
```

### Data Fields

- char [ringport](#) [16]  
*Porto do servidor TCP.*
- int [family](#)  
*Família da socket.*
- struct sockaddr \* [destination](#)  
*Estrutura de destino para as mensagens UDP.*
- socklen\_t [dest\\_size](#)  
*Tamanho da estrutura de destino para as mensagens UDP.*

### 3.3.1 Field Documentation

3.3.1.1 socklen\_t startup\_data::dest\_size

3.3.1.2 struct sockaddr\* startup\_data::destination

3.3.1.3 int startup\_data::family

3.3.1.4 char startup\_data::ringport[16]

The documentation for this struct was generated from the following file:

- include/[common.h](#)

## 3.4 transversal\_data Struct Reference

Estrutura com as referências necessárias aos vários módulos do projecto.

```
#include <common.h>
```

### Data Fields

- int [u](#)  
*Socket UDP.*
- int [t](#)  
*Socket TCP [accept].*
- struct [peer\\_data](#) [peer\\_pred](#)  
*Estrutura do predecessor.*
- struct [peer\\_data](#) [peer\\_succ](#)  
*Estrutura do sucessor.*
- int [ring](#)  
*Anel a ser utilizado.*
- int [id](#)  
*Identificador actual do nó.*
- char [reg](#)  
*Registered Node.*
- struct [startup\\_data](#) [startup\\_data](#)  
*Parâmetros da linha de comandos.*
- char [ext\\_addr](#) [40]  
*IP externo do servidor.*
- int [serv\\_arranq](#)
- int [socket\\_with\\_new\\_node](#)

### 3.4.1 Field Documentation

3.4.1.1 char transversal\_data::ext\_addr[40]

3.4.1.2 int transversal\_data::id

3.4.1.3 struct peer\_data transversal\_data::peer\_pred

3.4.1.4 struct peer\_data transversal\_data::peer\_succ

3.4.1.5 char transversal\_data::reg

3.4.1.6 int transversal\_data::ring

3.4.1.7 int transversal\_data::serv\_arranq

3.4.1.8 int transversal\_data::socket\_with\_new\_node

3.4.1.9 struct startup\_data transversal\_data::startup\_data

3.4.1.10 int transversal\_data::t

3.4.1.11 int transversal\_data::u

The documentation for this struct was generated from the following file:

- `include/common.h`

## Chapter 4

# File Documentation

### 4.1 include/common.h File Reference

Cabeçalho com declarações das estruturas de dados transversais aos vários módulos do projecto.

```
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/ip.h>
```

#### Data Structures

- struct [startup\\_data](#)  
*Estrutura com os dados a ser recolhidos pela linha de comandos.*
- struct [peer\\_data](#)  
*Estrutura com a informação referente a um par.*
- struct [transversal\\_data](#)  
*Estrutura com as referências necessárias aos vários módulos do projecto.*

### 4.2 include/interface.h File Reference

Cabeçalhos e protótipos associados às funções relacionadas com a interface com o utilizador.

#### Functions

- void [print\\_ui](#) ()  
*Função de impressão dos comandos disponíveis.*
- int [interface](#) (struct [transversal\\_data](#) \*[transversal\\_data](#))  
*Função de análise dos comandos inseridos.*

#### 4.2.1 Function Documentation

##### 4.2.1.1 int interface ( struct transversal\_data \* transversal\_data )

Após detecção de dados para ser lidos via stdin, esta função processa esses dados de maneira a executar o comando inserido pelo utilizador.

#### 4.2.1.2 void print\_ui ( )

Esta função mostra os comandos disponíveis para o utilizador. Os comandos sem parâmetros são hardcoded, enquanto que os comandos com parâmetros estão descritos em funções individuais de maneira a facilitar as mensagens de erro de formato.

### 4.3 include/net\_common.h File Reference

Cabeçalhos e protótipos associados às funções relacionadas com a rede mas não associadas a nenhum protocolo em específico.

#### Functions

- void [getsockaddr](#) (char \*node, char \*service, int \*family, socklen\_t \*size, struct sockaddr \*\*sockaddress, int protocol)

*Função de leitura de endereços e portos e criação de estruturas de destino.*

#### 4.3.1 Function Documentation

##### 4.3.1.1 void getsockaddr ( char \* node, char \* service, int \* family, socklen\_t \* size, struct sockaddr \*\* sockaddress, int protocol )

Esta função recebe um endereço e um porto, bem como a família (que é tanto uma entrada como uma saída), o tamanho da estrutura de saída e um ponteiro para a estrutura de saída. Caso esse ponteiro seja NULL, a estrutura é alocada.

A função getaddrinfo() é utilizada em vez da função gethostbyname(), por uma questão de preferência pessoal e também pelo facto da função getaddrinfo() estar especificada como thread-safe (a função gethostbyname() só é thread-safe em algumas implementações). Os detalhes da função getaddrinfo() podem ser consultados em [RFC 2553](#).

### 4.4 include/net\_tcp.h File Reference

Cabeçalhos e protótipos associados às funções relacionadas com o protocolo TCP.

#### Functions

- void [createserver\\_tcp](#) (struct transversal\_data \*transversal\_data)

*Função de abertura do servidor TCP.*

- int [connect\\_tcp](#) (char \*node, char \*service)

*Função de abertura de uma sessão TCP com outro nó.*

#### 4.4.1 Function Documentation

##### 4.4.1.1 int connect\_tcp ( char \* node, char \* service )

Esta função cria uma socket e liga-a directamente ao nó, retornando então essa mesma socket.



#### 4.4.1.2 void createserver\_tcp ( struct transversal\_data \* transversal\_data )

Esta função cria a socket para recepção de mensagens via TCP. Esta função consiste de socket(), bind() e listen() exclusivamente, o accept deve ser feito a posteriori, de maneira a não bloquear a execução do programa.

## 4.5 include/net\_udp.h File Reference

Cabeçalhos e protótipos associados às funções relacionadas com o protocolo UDP.

### Functions

- void [getudpdest](#) (char \*node, char \*service, struct [startup\\_data](#) \*startup\_data)

*Função de determinação do destino do protocolo UDP.*

- void [createsocket\\_udp](#) (struct [transversal\\_data](#) \*transversal\_data)

*Função de abertura da socket UDP com o servidor de arranque.*

### 4.5.1 Function Documentation

#### 4.5.1.1 void createsocket\_udp ( struct transversal\_data \* transversal\_data )

Esta função recebe como parâmetros a estrutura de dados transversal ao programa e abre uma socket UDP tendo como destino o servidor de arranque.

#### 4.5.1.2 void getudpdest ( char \* node, char \* service, struct startup\_data \* startup\_data )

Esta função recebe o endereço e porto do servidor de arranque e coloca na estrutura comum a estrutura correspondente a esse endereço.

Na estrutura comum é alocada memória para a struct sockaddr de destino.

## 4.6 include/options.h File Reference

Cabeçalhos e protótipos associados ao módulo de processamento das opções de linha de comandos.

```
#include "common.h"
```

### Data Structures

- struct [args\\_parse\\_options](#)

*Estrutura com os argumentos para a função [parse\\_options](#).*

### Functions

- void \* [parse\\_options](#) (struct [args\\_parse\\_options](#) \*params)

*Função responsável pela interpretação dos argumentos passados via linha de comandos.*

### 4.6.1 Function Documentation

#### 4.6.1.1 void\* parse\_options ( struct args\_parse\_options \* *params* )

Esta função recorre à função getopt() de maneira a obter os diversos parâmetros passados via linha de comandos. Os argumentos são passados via ponteiro para estrutura de maneira a que a função seja compatível com threads.

## 4.7 include/ring.h File Reference

```
#include "common.h"
```

### Functions

- int [join\\_ring](#) (char \*ring, char \*num, struct [transversal\\_data](#) \*transversal\_data)

### 4.7.1 Function Documentation

#### 4.7.1.1 int join\_ring ( char \* *ring*, char \* *num*, struct transversal\_data \* *transversal\_data* )

## 4.8 include/trata\_message.h File Reference

### Functions

- int [check\\_message](#) (char \*\*message, int num\_words)
- void [write\\_message\\_tcp](#) (char \*string, int socket)
- void [trata\\_mensagem](#) (char \*buffer)
- void [preenche\\_predi\\_info](#) (struct [transversal\\_data](#) \*transversal\_data, char \*id, char \*ip, char \*porto)
- int [dist](#) (int ele, int eu)
- int [verifica\\_se\\_responsavel](#) (char \*c, int eu\_id, int predi\_id)

### 4.8.1 Function Documentation

#### 4.8.1.1 int check\_message ( char \*\* *message*, int *num\_words* )

#### 4.8.1.2 int dist ( int *ele*, int *eu* )

#### 4.8.1.3 void preenche\_predi\_info ( struct transversal\_data \* *transversal\_data*, char \* *id*, char \* *ip*, char \* *porto* )

#### 4.8.1.4 void trata\_mensagem ( char \* *buffer* )

#### 4.8.1.5 int verifica\_se\_responsavel ( char \* *c*, int *eu\_id*, int *predi\_id* )

#### 4.8.1.6 void write\_message\_tcp ( char \* *string*, int *socket* )

## 4.9 src/interface.c File Reference

Interface com o utilizador.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include "common.h"
#include "ring.h"
#include "trata_message.h"
```

## Macros

- `#define _XOPEN_SOURCE 700`

## Functions

- void `print_join_s` ()  
*Função de impressão de formato do comando <join> (curto).*
- void `print_join_l` ()  
*Função de impressão de formato do comando <join> (longo).*
- void `print_search` ()  
*Função de impressão de formato do comando <search>.*
- void `print_ui` ()  
*Função de impressão dos comandos disponíveis.*
- void `print_error` ()  
*Função de impressão de erro nos comandos.*
- int `interface` (struct `transversal_data` \*`transversal_data`)  
*Função de análise dos comandos inseridos.*

### 4.9.1 Detailed Description

Este ficheiro contém as funções relacionadas com a interface com o utilizador.

### 4.9.2 Macro Definition Documentation

#### 4.9.2.1 `#define _XOPEN_SOURCE 700`

### 4.9.3 Function Documentation

#### 4.9.3.1 `int interface ( struct transversal_data * transversal_data )`

Após detecção de dados para ser lidos via stdin, esta função processa esses dados de maneira a executar o comando inserido pelo utilizador.

#### 4.9.3.2 `void print_error ( )`

Esta função mostra uma mensagem de erro, informando o utilizador que o comando que inseriu é inválido.

4.9.3.3 void print\_join\_l ( )

4.9.3.4 void print\_join\_s ( )

4.9.3.5 void print\_search ( )

4.9.3.6 void print\_ui ( )

Esta função mostra os comandos disponíveis para o utilizador. Os comandos sem parâmetros são hardcoded, enquanto que os comandos com parâmetros estão descritos em funções individuais de maneira a facilitar as mensagens de erro de formato.

## 4.10 src/main.c File Reference

Ficheiro principal do projecto.

```
#include <stdlib.h>
#include <stdio.h>
#include "common.h"
#include "options.h"
#include "net_udp.h"
#include "net_tcp.h"
#include "interface.h"
```

### Functions

- int [main](#) (int argc, char \*\*argv)

*Função principal do programa ddt.*

#### 4.10.1 Detailed Description

Este ficheiro serve como chave de abóbada para os vários módulos desenvolvidos e responsáveis pelas diversas funcionalidades do projecto.

#### 4.10.2 Function Documentation

4.10.2.1 int main ( int *argc*, char \*\* *argv* )

Esta função serve-se dos módulos criados nos restantes ficheiros-fonte para implementar as funcionalidades necessárias ao projecto, como especificadas no enunciado.

## 4.11 src/net\_common.c File Reference

Ficheiro responsável pela implementação das funções relacionadas com rede mas não associadas a nenhum dos protocolos utilizados.

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netdb.h>
#include <string.h>
#include "common.h"
#include "net_common.h"
```

## Macros

- `#define _XOPEN_SOURCE`

## Functions

- void `getsockaddr` (char \*node, char \*service, int \*family, socklen\_t \*size, struct sockaddr \*\*sockaddress, int protocol)

*Função de leitura de endereços e portos e criação de estruturas de destino.*

### 4.11.1 Macro Definition Documentation

#### 4.11.1.1 `#define _XOPEN_SOURCE`

### 4.11.2 Function Documentation

#### 4.11.2.1 void `getsockaddr` ( char \* node, char \* service, int \* family, socklen\_t \* size, struct sockaddr \*\* sockaddress, int protocol )

Esta função recebe um endereço e um porto, bem como a família (que é tanto uma entrada como uma saída), o tamanho da estrutura de saída e um ponteiro para a estrutura de saída. Caso esse ponteiro seja NULL, a estrutura é alocada.

A função `getaddrinfo()` é utilizada em vez da função `gethostbyname()`, por uma questão de preferência pessoal e também pelo facto da função `getaddrinfo()` estar especificada como thread-safe (a função `gethostbyname()` só é thread-safe em algumas implementações). Os detalhes da função `getaddrinfo()` podem ser consultados em [RFC 2553](#).

## 4.12 src/net\_tcp.c File Reference

Ficheiro responsável pela implementação das funções de rede associadas ao protocolo TCP.

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include "common.h"
#include "net_common.h"
#include "net_tcp.h"
```

## Macros

- `#define RCI_BACKLOG 16`

## Functions

- void `createserver_tcp` (struct `transversal_data` \*`transversal_data`)  
*Função de abertura do servidor TCP.*
- int `connect_tcp` (char \*`node`, char \*`service`)  
*Função de abertura de uma sessão TCP com outro nó.*

### 4.12.1 Macro Definition Documentation

#### 4.12.1.1 `#define RCI_BACKLOG 16`

### 4.12.2 Function Documentation

#### 4.12.2.1 int `connect_tcp` ( char \* *node*, char \* *service* )

Esta função cria uma socket e liga-a directamente ao nó, retornando então essa mesma socket.

#### 4.12.2.2 void `createserver_tcp` ( struct `transversal_data` \* *transversal\_data* )

Esta função cria a socket para recepção de mensagens via TCP. Esta função consiste de `socket()`, `bind()` e `listen()` exclusivamente, o `accept` deve ser feito a posteriori, de maneira a não bloquear a execução do programa.

## 4.13 `src/net_udp.c` File Reference

Ficheiro responsável pela implementação das funções de rede associadas ao protocolo UDP.

```
#include <stdlib.h>
#include <stdio.h>
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netdb.h>
#include <string.h>
#include <stdint.h>
#include <arpa/inet.h>
#include "common.h"
#include "net_common.h"
#include "net_udp.h"
```

## Functions

- void `getudpdest` (char \*`node`, char \*`service`, struct `startup_data` \*`startup_data`)  
*Função de determinação do destino do protocolo UDP.*
- void `createsocket_udp` (struct `transversal_data` \*`transversal_data`)  
*Função de abertura da socket UDP com o servidor de arranque.*

### 4.13.1 Function Documentation

#### 4.13.1.1 void `createsocket_udp` ( struct `transversal_data` \* *transversal\_data* )

Esta função recebe como parâmetros a estrutura de dados transversal ao programa e abre uma socket UDP tendo como destino o servidor de arranque.

#### 4.13.1.2 void getudpdest ( char \* node, char \* service, struct startup\_data \* startup\_data )

Esta função recebe o endereço e porto do servidor de arranque e coloca na estrutura comum a estrutura correspondente a esse endereço.

Na estrutura comum é alocada memória para a struct sockaddr de destino.

## 4.14 src/options.c File Reference

Ficheiro responsável pela interpretação dos argumentos passados via linha de comandos.

```
#include <unistd.h>
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#include "common.h"
#include "options.h"
#include "net_udp.h"
```

### Macros

- `#define \_XOPEN\_SOURCE`

### Functions

- void [usage](#) (char \*appname)  
*Função que mostra a mensagem Usage.*
- void \* [parse\\_options](#) (struct [args\\_parse\\_options](#) \*params)  
*Função responsável pela interpretação dos argumentos passados via linha de comandos.*

#### 4.14.1 Detailed Description

As funções implementadas neste módulo correspondem à interpretação dos argumentos passados via linha de comandos, e correspondente transferência para a estrutura de dados transversal aos vários módulos do projecto.

#### 4.14.2 Macro Definition Documentation

##### 4.14.2.1 `#define \_XOPEN\_SOURCE`

#### 4.14.3 Function Documentation

##### 4.14.3.1 void\* [parse\\_options](#) ( struct [args\\_parse\\_options](#) \* *params* )

Esta função recorre à função getopt() de maneira a obter os diversos parâmetros passados via linha de comandos. Os argumentos são passados via ponteiro para estrutura de maneira a que a função seja compatível com threads.

##### 4.14.3.2 void [usage](#) ( char \* *appname* )

## 4.15 src/ring.c File Reference

```
#include <stdio.h>
#include <string.h>
#include <sys/types.h>
#include <sys/socket.h>
#include "ring.h"
#include "common.h"
```

### Macros

- `#define RCI_MSGSIZE 256`

### Functions

- `int join_ring (char *ring, char *num, struct transversal_data *transversal_data)`

#### 4.15.1 Macro Definition Documentation

4.15.1.1 `#define RCI_MSGSIZE 256`

#### 4.15.2 Function Documentation

4.15.2.1 `int join_ring ( char * ring, char * num, struct transversal_data * transversal_data )`

## 4.16 src/trata\_message.c File Reference

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "common.h"
#include "trata_message.h"
#include "math.h"
```

### Functions

- `int dist (int ele, int eu)`
- `int verifica_se_responsavel (char *c, eu_id, predi_id)`
- `int check_message (char **message, int num_words)`
- `void write_message_tcp (char *string, int socket)`
- `void preenche_predi_info (struct transversal_data *transversal_data, char *id, char *ip, char *porto)`
- `void trata_mensagem (char *buffer, struct transversal_data *transversal_data)`

#### 4.16.1 Function Documentation

4.16.1.1 `int check_message ( char ** message, int num_words )`

4.16.1.2 `int dist ( int ele, int eu )`



4.16.1.3 void preenche\_predi\_info ( struct transversal\_data \* *transversal\_data*, char \* *id*, char \* *ip*, char \* *porto* )

4.16.1.4 void trata\_mensagem ( char \* *buffer*, struct transversal\_data \* *transversal\_data* )

4.16.1.5 int verifica\_se\_responsavel ( char \* *c*, eu\_id , predi\_id )

4.16.1.6 void write\_message\_tcp ( char \* *string*, int *socket* )



# Index

## `_XOPEN_SOURCE`

interface.c, [13](#)  
net\_common.c, [15](#)  
options.c, [17](#)

## `argc`

args\_parse\_options, [5](#)

## `args_parse_options`

argc, [5](#)  
argv, [5](#)  
startup\_data, [5](#)

## `argv`

args\_parse\_options, [5](#)

## `check_message`

trata\_message.c, [18](#)  
trata\_message.h, [12](#)

## `connect_tcp`

net\_tcp.c, [16](#)  
net\_tcp.h, [10](#)

## `createserver_tcp`

net\_tcp.c, [16](#)  
net\_tcp.h, [10](#)

## `createsocket_udp`

net\_udp.c, [16](#)  
net\_udp.h, [11](#)

## `dest_size`

startup\_data, [6](#)

## `destination`

startup\_data, [6](#)

## `dist`

trata\_message.c, [18](#)  
trata\_message.h, [12](#)

## `ext_addr`

transversal\_data, [7](#)

## `family`

startup\_data, [6](#)

## `getsockaddr`

net\_common.c, [15](#)  
net\_common.h, [10](#)

## `getudpdest`

net\_udp.c, [16](#)  
net\_udp.h, [11](#)

## `id`

peer\_data, [6](#)  
transversal\_data, [7](#)

include/common.h, [9](#)

include/interface.h, [9](#)

include/net\_common.h, [10](#)

include/net\_tcp.h, [10](#)

include/net\_udp.h, [11](#)

include/options.h, [11](#)

include/ring.h, [12](#)

include/trata\_message.h, [12](#)

## `interface`

interface.c, [13](#)

interface.h, [9](#)

## `interface.c`

`_XOPEN_SOURCE`, [13](#)

interface, [13](#)

print\_error, [13](#)

print\_join\_l, [13](#)

print\_join\_s, [14](#)

print\_search, [14](#)

print\_ui, [14](#)

## `interface.h`

interface, [9](#)

print\_ui, [9](#)

## `join_ring`

ring.c, [18](#)

ring.h, [12](#)

## `main`

main.c, [14](#)

## `main.c`

main, [14](#)

## `net_common.c`

`_XOPEN_SOURCE`, [15](#)

getsockaddr, [15](#)

## `net_common.h`

getsockaddr, [10](#)

## `net_tcp.c`

connect\_tcp, [16](#)

createserver\_tcp, [16](#)

RCI\_BACKLOG, [16](#)

## `net_tcp.h`

connect\_tcp, [10](#)

createserver\_tcp, [10](#)

## `net_udp.c`

createsocket\_udp, [16](#)

getudpdest, [16](#)

## `net_udp.h`

createsocket\_udp, [11](#)

getudpdest, [11](#)

- node
  - peer\_data, 6
- options.c
  - \_XOPEN\_SOURCE, 17
  - parse\_options, 17
  - usage, 17
- options.h
  - parse\_options, 12
- parse\_options
  - options.c, 17
  - options.h, 12
- peer\_data, 5
  - id, 6
  - node, 6
  - service, 6
  - socket, 6
- peer\_pred
  - transversal\_data, 7
- peer\_succ
  - transversal\_data, 7
- preenche\_predi\_info
  - trata\_message.c, 18
  - trata\_message.h, 12
- print\_error
  - interface.c, 13
- print\_join\_l
  - interface.c, 13
- print\_join\_s
  - interface.c, 14
- print\_search
  - interface.c, 14
- print\_ui
  - interface.c, 14
  - interface.h, 9
- RCI\_BACKLOG
  - net\_tcp.c, 16
- RCI\_MSGSIZE
  - ring.c, 18
- reg
  - transversal\_data, 7
- ring
  - transversal\_data, 7
- ring.c
  - join\_ring, 18
  - RCI\_MSGSIZE, 18
- ring.h
  - join\_ring, 12
- ringport
  - startup\_data, 6
- serv\_arranq
  - transversal\_data, 7
- service
  - peer\_data, 6
- socket
  - peer\_data, 6
- socket\_with\_new\_node
  - transversal\_data, 7
- src/interface.c, 12
- src/main.c, 14
- src/net\_common.c, 14
- src/net\_tcp.c, 15
- src/net\_udp.c, 16
- src/options.c, 17
- src/ring.c, 18
- src/trata\_message.c, 18
- startup\_data, 6
  - args\_parse\_options, 5
  - dest\_size, 6
  - destination, 6
  - family, 6
  - ringport, 6
  - transversal\_data, 7
- t
  - transversal\_data, 7
- transversal\_data, 7
  - ext\_addr, 7
  - id, 7
  - peer\_pred, 7
  - peer\_succ, 7
  - reg, 7
  - ring, 7
  - serv\_arranq, 7
  - socket\_with\_new\_node, 7
  - startup\_data, 7
  - t, 7
  - u, 7
- trata\_message.c
  - check\_message, 18
  - dist, 18
  - preenche\_predi\_info, 18
  - trata\_mensagem, 19
  - verifica\_se\_responsavel, 19
  - write\_message\_tcp, 19
- trata\_message.h
  - check\_message, 12
  - dist, 12
  - preenche\_predi\_info, 12
  - trata\_mensagem, 12
  - verifica\_se\_responsavel, 12
  - write\_message\_tcp, 12
- trata\_mensagem
  - trata\_message.c, 19
  - trata\_message.h, 12
- u
  - transversal\_data, 7
- usage
  - options.c, 17
- verifica\_se\_responsavel
  - trata\_message.c, 19
  - trata\_message.h, 12

write\_message\_tcp  
    trata\_message.c, [19](#)  
    trata\_message.h, [12](#)