1. Node结构体

|  |
| --- |
| type Node struct {  IP net.IP // len 4 for IPv4 of 16 for IPv6  UDP uint16 // UDP port number  TCP uint16 // TCP port number  ID NodeID // the node's public key (ECC public key marshalled)  sha common.Hash //NodeID的哈希值  } |

1. 函数

|  |
| --- |
| func newNode(id NodeID, ip net.IP, udpPort, tcpPort uint16) \*Node  新建一个Node类型的节点 |

|  |
| --- |
| func ParseNode(rawurl string) (\*Node, error)  对字符串形式的url进行解析，将提取的信息进行组合构成Node结构体  (取出各部分：NodeID , IP , UDP Port , TCP Port) |

|  |
| --- |
| func MustParseNode(rawurl string) \*Node  底层调用了ParseNode()函数，也是对url进行解析。与ParseNode()不同的是增加了解析出错发出panic警报的功能 |

|  |
| --- |
| func HexID(in string) (NodeID, error)  将一个表示十六进制数字的字符串转换为NodeID (字符串可以以0x开头) |

|  |
| --- |
| func MustHexID(in string) NodeID  底层调用了HexID()，增加了转换失败时发出panic警报 |

|  |
| --- |
| func PubkeyID(pub \*ecdsa.PublicKey) NodeID  将传入的公钥采用椭圆曲线算法序列化后得到NodeID |

|  |
| --- |
| func recoverNodeID(hash, sig []byte) (id NodeID, err error)  通过数字签名与获取消息的哈希计算出签名者的公钥 |

|  |
| --- |
| func distcmp(target, a, b common.Hash) int  分别比较目标target与a和b的距离。如果a距离目标更近就返回-1 如果b距离目标更近就返回1 如果距离相等就返回0。  距离的计算是通过异或比较哈希值，异或之后的结果越大说明距离target越远  需要注意：common.Hash是byte数组，因此每次异或比较都是对每一个元素byte进行比较(而不是以bit为单位进行比较)  从比较的方式来看，common.Hash是小端字节序。common.Hash[0]存储最高位的哈希值 |

|  |
| --- |
| func logdist(a, b common.Hash) int  返回a与b的对数距离 --> log2(a ^ b) 。 这个距离是指异或后的前缀0的个数  返回值是a与b不相等的bit位数(异或后去除前缀0的剩余bit位数) |

|  |
| --- |
| func hashAtDistance(a common.Hash, n int) (b common.Hash)  本函数返回一个满足 logdist(a, b) == n 的随机哈希值 b (哈希值a与距离n是已知的) |

1. Node结构体的方法

|  |
| --- |
| func (n NodeID) String() string  以字符串形式返回NodeID |

|  |
| --- |
| func (n \*Node) Sha()  根据节点的ID值计算其哈希值，赋给n.sha |

|  |
| --- |
| func (n \*Node) addr() \*net.UDPAddr  获取节点的UDP地址 |

|  |
| --- |
| func (n \*Node) String() string  以字符串形式返回节点n的url信息  url格式：scheme://userinfo@host/path?query#fragment  本函数的返回格式为：dpnode://User@Host?discport=n.UDP |

1. NodeID类型变量的方法

type NodeID [nodeIDBits / 8]byte

|  |
| --- |
| func (id NodeID) Pubkey() (\*ecdsa.PublicKey, error)  用NodeID创建一个椭圆曲线结构体公钥 |