int

bytes 0: 4

type | primitive_type

level | 值

1~4 5~8

type=0
primitive_type=0
vector_type=-1;
level = 读(1-4)
count=1;
length = intBytes=4;
size=length+headerBytes
+intBytes=9
intData = 读(5-8)

bool

bytes 0 → 值

type | primitive_type

1-4

level

type=0
primitive_type=0
vector_type=-1;
level = 读(1-4)
count=1
length = boolBytes=0;
size=length+headerBytes;
+intBytes=5

boolData = 读(0→0)

double

byte 0

type | primitive_type

level | 值

1-4 5~12

type=0
primitive_type=0
vector_type=-1
level=读(1-4)
count=1;
length = doubleBytes=8;
size=length+headerBytes
+intBytes=13
doubleData = 读(5~12)

String

byte 0

type | primitive_type

level | length | 值

1~4 5~8 ~ 长为length

type=0
primitive_type=0
Vector_type=-1
level = 读(1-4)
length = 读(5-8)
size=length+headerBytes
+intBytes*2;
StringData = 读(从9开始, 长为length)

vector

byte 0

type | vector_type

level | count

1-4 5~8

值域

SendableData 共有count个

type=1
primitive_type=-1
vector_type = 读(0→0)
level = 读1-4
count = 读5-8
length = 读值域, 统计size总值
size = length+headerBytes+intBytes*2
VectorData = 读值域

pairVector

byte 0

type | vector_type

level | count

值域

SendableData 共有count对(count*2个)

type=2
primitive_type=-1
Vector_type = 读(0→0)
level = 读1-4
count = 读5-8
length = 读值域, 统计size总值
size = length+headerBytes+intBytes*2;
pairVectorData = 读值域

constructor:
(int, level)
(bool, level)
(double, level)
(string, length, level)
(vector<int>, level) 3种
(vector<string>, length, level)
(vector<自己>, se..type, level)
(vector<pair... - - -)

retriver:
- getInt()
- getBool()
- getDouble()
- getString()
- getVector()
- getPairVector()
- getItem(index)
- getPairItem(index)

加解密:
static void setFail(code(char*))
char* encode()
decode(char*)

meta:
信息外套一层:

check

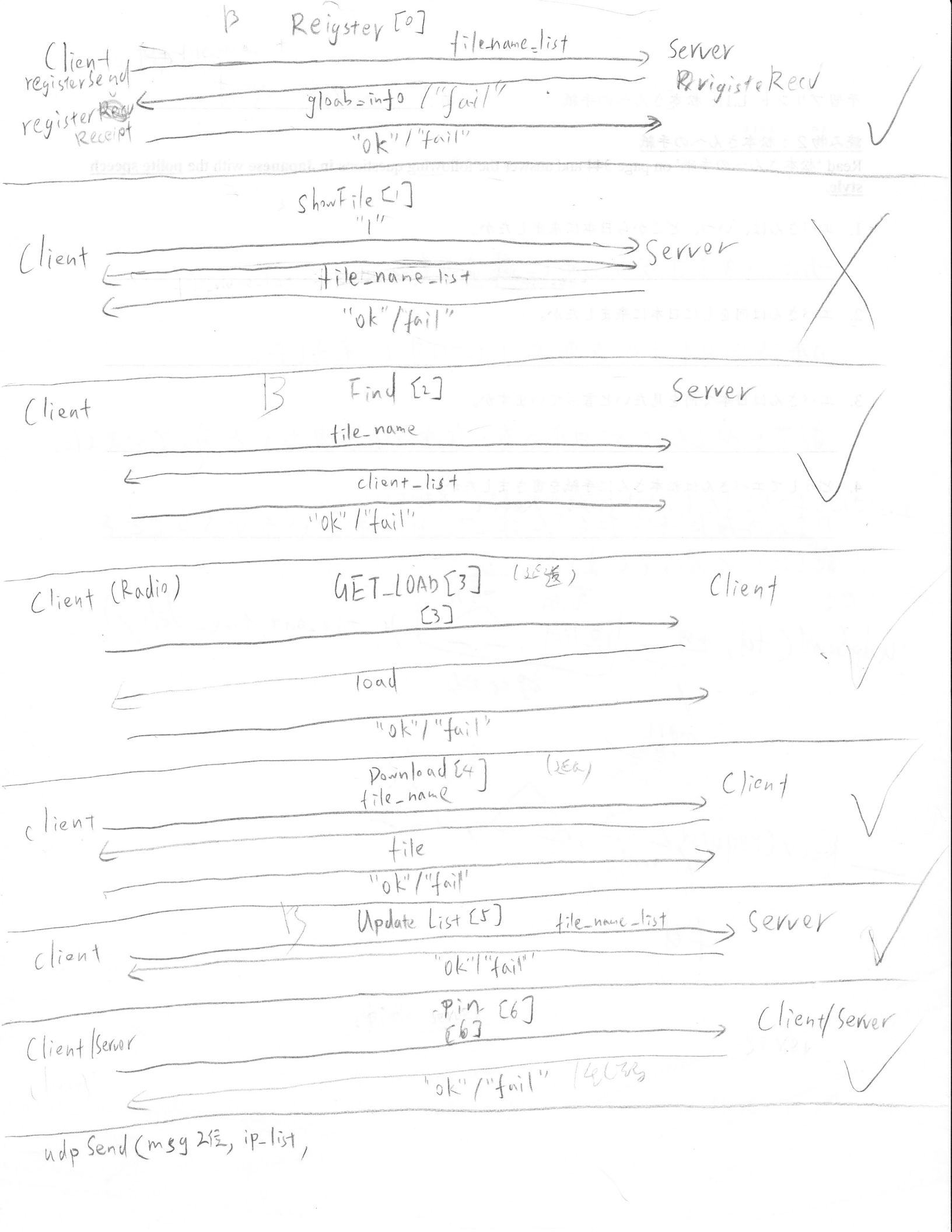
总长(含自己) | STATE

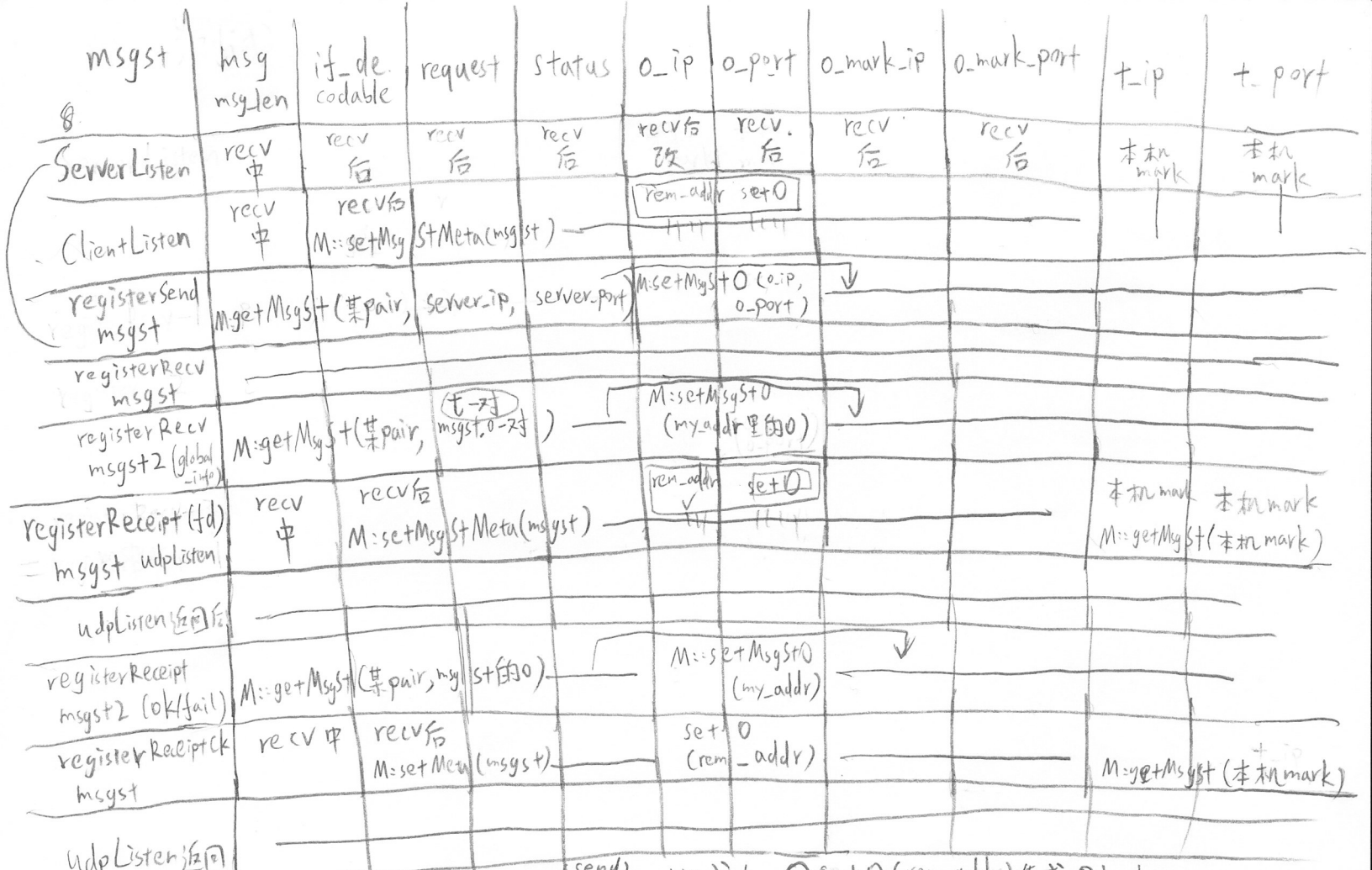
0: X
1: V

R

checksum | S

m_i:0 | m_par | m_id


Register [0]
Client
registerSend
registerRecv
Receipt
file_name_list
Server
RegisteRecv
gloab_info ("fail"
"ok" / "fail"
ShowFile [1]
"1"
Client
Server
file_name_list
"ok" / fail"
Find [2]
Client
Server
file_name
client_list
"ok" / "fail"
Client (Radio)
GET_LOAD [3]
[3]
load
"ok" / "fail"
Client
Download [4]
file_name
client
file
"ok" / "fail"
Client
Update List [5]
file_name_list
Server
client
"ok" / "fail"
Pin [6]
[6]
Client/Server
Client/Server
"ok" / "fail"
udp Send (msg



① getMsgSt(send)(本机mark)生成t ④set0(rem_addr)生成0
②recv生成msg, ③M::setMeta生成 if/req/status/mark C

ServerListen / registerRecv → msgst 接收用 → 传入registerSend / 从udpListen返回后信息在

registerRecv → msgst2 发送用 ① getMsgSt(pair, msgst的0 ⇒ msgst的t) ② set0(my_addr) (recv sendto前, udpSend里完成)

registerReceipt udpListen → msgst 接 → get(本机mark); recv生成msg, setMeta, set0

registerReceipt → msgst2 发 ① getMsgSt(pair, st的0) ② set0(my_addr)

① 如果msgst不decodable 传回FAIL
② 如果FAIL, 不动了。
③ 如果ok, 包装msgst2
④ 如果msgst2是FAIL, 不管了

msgst.if_decodable != true || msgst.status == Ok → 传msgst2 → 值 / FAIL

MsgSt类型: Ok ✓ Fail ✓ FileName ✓ FileNameList ✓ GlobalInfo ✓

File ✓ ClientList ✓ printMsgStMeta(st) ✓

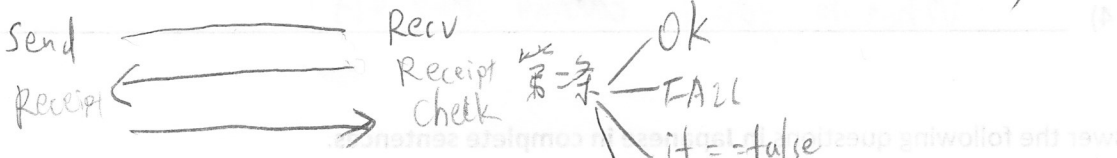
Todo

1. 完整测试 MsgSt / Message

getMsgSt Recv(t_ip, t_port)

getMsgSt Send (

2. 完整测试路径第一条



delay_max 改成 time_out_max

MsgSt测试

发送用 file_nam_list	Ok	有Noise
FileName	✓	✓
Ok	✓	✓
FAIL	✓	✓
FileNameList	✓	✓
GlobalInfo	✓	✓
FILE	✓	✓
ClientList	✓	✓