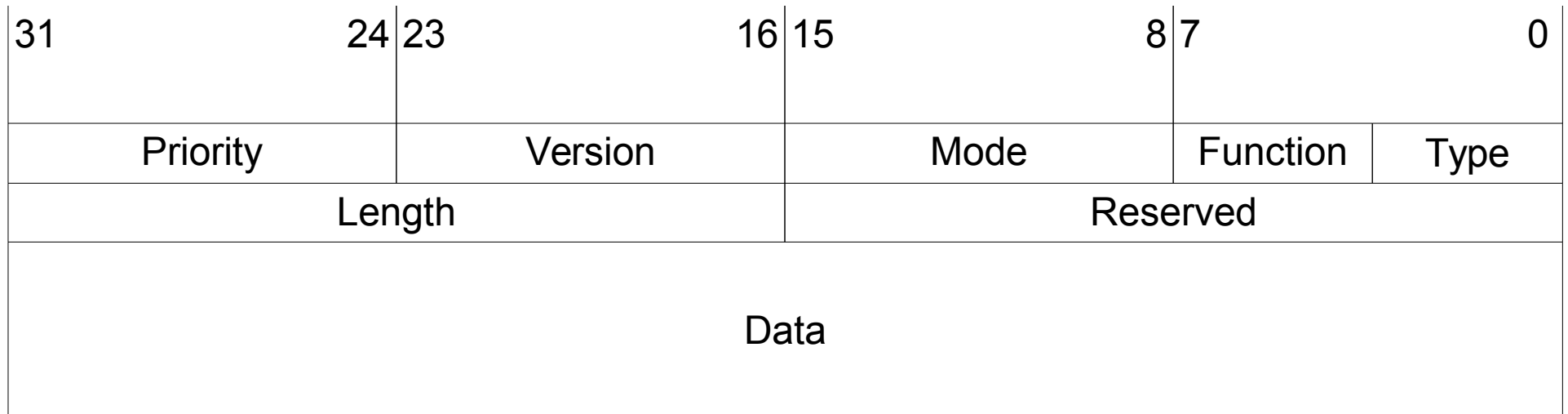


Labor Verteilte Systeme

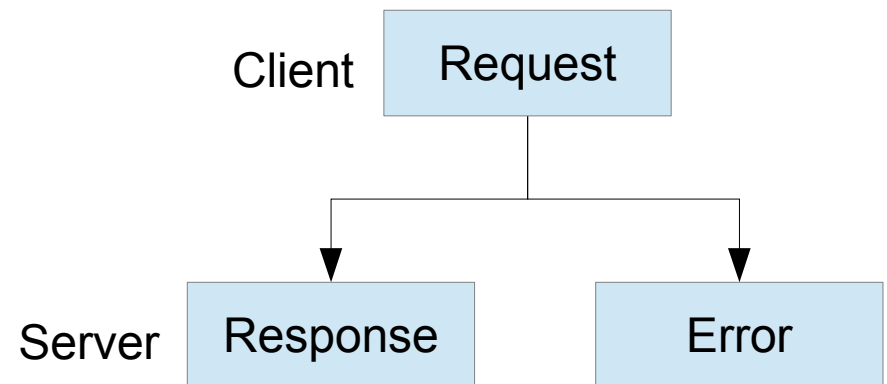
Protokoll Entwurf V2.6
MES Semester 2
10.5.2016

Nachrichten Header V2.6



Key features

- Kompatibel zur Status Anfrage
- Einfache Umsetzung



- Für alle Labore einsetzbar
 - Labor 1: Priorität beliebig belegen
 - Labor 1 & 2: Keine Broadcasts unterstützen
- Einfach Aktualisierbar
- Client und Server Port: 11111

Beschreibung der Felder

- Priority : Nachrichten Priorität $0 > 255$
- Version : Protokollversion
PROTOCOL_VERSION = 14
- Sender Type / Mode:
 - 1 : Status Tracker MODE_STATUS
 - 2 : Server MODE_SERVER
 - 3 : Client MODE_CLIENT
- Length : Länge der Paketdaten (ohne Header)

Beschreibung der Felder

- Function (4 bit):
 - 0 : Set Polynome
FNC_POLYNOME
 - 1 : Decrypt Data
FNC_DECRYPT
 - 2 : Unlock server
FNC_UNLOCK
 - 5 : Broadcast
FNC_BROADCAST
 - 6 : Status
FNC_STATUS
- Type (4 bit):
 - 3 : Request
MSG_REQUEST
 - 4 : Response
MSG_RESPONSE
 - 15 : Error
MSG_ERROR

Fehlercodes

- 0 : No Error
NO_ERROR
- 1 : Wrong Length
ERR_PACKETLENGTH
- 2 : Invalid Version
ERR_INVALIDVERSION
- 3 : Invalid Mode
ERR_INVALIDMODE
- 4 : Unknown Function
ERR_NOSUCHFUNCTION
- 5 : Invalid Type
ERR_INVALIDTYPE
- 6 : Inkonsistenter Paketheader
ERR_HEADER_DATA
- 8 : Data Error
ERR_DATA
- 16 : Server already used
ERR_SERVERINUSE
- 32: Function timed out
ERR_FUNCTIONTIMEOUT
- 33 : Funktions Ausführungs Fehler
ERR_FUNCTIONEXEC
- 64 : Entschlüsselungs Fehler
ERR_DECRYPT
- 128 : Allocation Error
ERR_ALLOC
- 129 : Invalid Pointer Error
ERR_INVALID_PTR
- 130 : Wrong Client / Server Connected
ERR_NOTFORME
- 131 : No Generator Polynome was set
ERR_NO_GP

Fehlercodes (2)

- 252 : Konnte Paket nicht senden

ERR_SEND_ERROR

- 253 : API nicht initialisiert

ERR_NO_INIT

- 254 : Kein Paket am Socket

ERR_NO_PACKET

- 255 : Unknown Error

ERR_UNKNOWN

Client-Server Kommunikation

- Set Polynome : 2 Bytes (**signed**) Client ID
16 Bit Polynome
 - Response : Empty
- Decrypt Data : 2 Bytes (**signed**) Client ID
16 Bit Block ID
Data to decrypt (n Bytes)
 - Response : 2 Bytes (**signed**) Client ID
2 Byte Block ID
Decrypted Data (n Bytes)

Client-Server Kommunikation

- Unlock Server : 2 Byte (**signed**) Client ID
 - Response : Empty

Broadcasts und Statusabfragen

- Broadcast : Empty
 - Response : Empty (Adresse vom Socket holen)
- Status : Empty
 - Response : 2 Bytes (**signed**) Client ID
 - 2 Reserved
 - 4 Bytes Sequence Number

Fehler Nachrichten

- Header wie empfangen
- Data : 1 Byte Error Code
2 Byte Block ID
- Block ID nur bei ERR_SERVERINUSE & ERR_DECRYPT sonst 0