

*Program structures for EPIC-
LITE experimental
implementation*

Linda Vidjak, Maja Štula, Julije Ožegović
{linda.vidjak, maja.stula, julije.ozegovic}@fesb.hr

Input data

compression

```
45 00 00 3c ee b2 40 00 40
45 00 00 34 ee b3 40 00 40
45 00 00 34 ee b4 40 00 40
45 08 00 34 ee b5 40 00 40
45 08 00 34 ee b6 40 00 40
45 08 00 34 ee b7 40 00 40
45 08 00 34 ee b8 40 00 40
45 08 00 34 ee b9 40 00 40
45 08 00 34 ee ba 40 00 40
45 08 00 34 ee bb 40 00 40
45 08 00 34 ee bc 40 00 40
45 08 00 34 ee bd 40 00 40
45 08 00 34 ee be 40 00 40
```



```
<E <- @ y0É ÁvÁ[ÁvÁrpóÑJ
: 0 obj<<□/Length 6601□/F
t%EWB□ÚÈ□~Bç Áo•ŠõÑü"t) Ê^ .i
□03P|QÁ4 ,fS-ò%`»&cÍ^Y6[9-Ä
t%,,WÛŽÛÈ□)Y~"G2 °dö...lòq□%□□
- 'æ-bÚ□...) I0£, iùüÈÄ~ÚZ?ä²
t%WKoÜF□%éW□%,,□d(v7 >M@`f_
òzÈ0×A >Y/Zn`€A□□>Ç"ñ`"éTè)
t%WMOÚÈ□%i~" >ÉÀ'Ép"¹9Ú□;ø
```



input data array

0100 0101
0000 0000
0000 0000
0011 1100
1110 1110
1101 0010
0100 0000
0000 0000
0100 0000
...
...
...

decompression

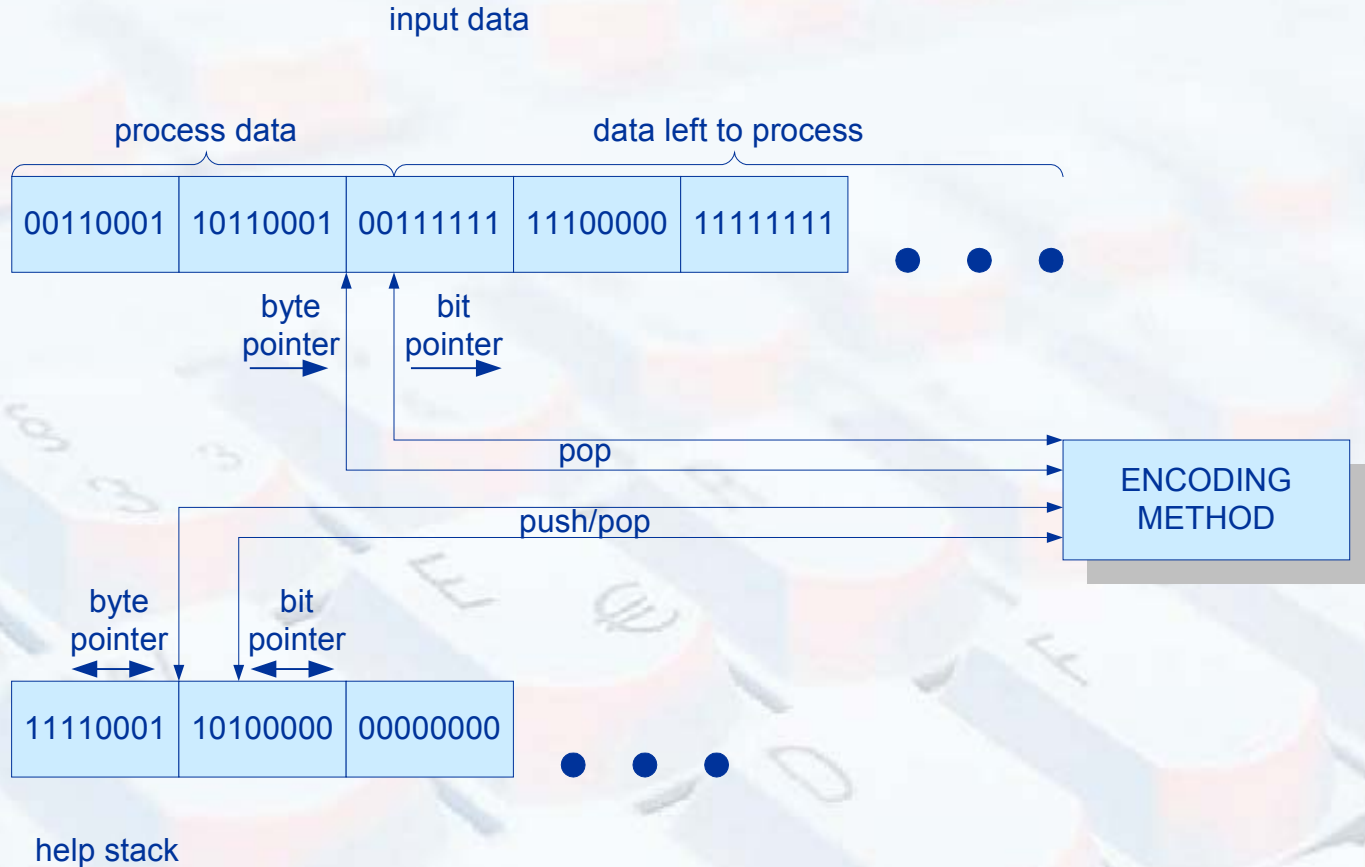
```
fc d6 98 08 a2 ac 92 dc 00
fc d9 78 08 c2 ac 92 dc 00
fc d1 58 08 e2 ac 92 dc 00
35 44 18 19 1a 1b 1c 1d 1e
10 4c 20 21 22 23 24 25 26
3b d4 28 29 2a 2b 2c 2d 2e
16 5c 30 31 32 33 34 35 36
3d 64 38 39 3a 3b 3c 3d 3e
12 ec 40 41 42 43 44 45 46
21 74 48 49 4a 4b 4c 4d 4e
2d fc 50 51 52 53 54 55 56
0f 05 58 59 5a 5b 5c 5d 5e
2a 0d 60 61 62 63 64 65 66
```



```
\ ùMr□€Öice@ 1
2 0 obj<<□/Length 6601
H%EWB□ÚÈ□~Bç Áo•ŠõÑü"t) Ê
!øx;LÜ.77□LcT±¤,L»O„S;
H%,,WÛŽÛÈ□)Y~"G2 °dö...lòq□
ZxÛ?ð/tB□' «□\C±ÈÄà+□W□
H%WKoÜF□%éW□%,,□d(v7 >M@`f_
„9EÄäï□f:ø□oJhîŠÇSnIB1□
H%WMOÚÈ□%i~" >ÉÀ'Ép"¹9Ú□;ø
```



Bit manipulation



○ Requirements for bit manipulation procedures:

- ◆ preserve original input data array
- ◆ append input data with compressor generated data.

Encoding methods

- *Elementary methods* – methods that process single header field (STATIC, VALUE, IRREGULAR, IRREGULAR-PADDED, INFERRED-IP-CHECKSUM, and LSB).
- *Complex methods:*
 - ◆ methods that put some additional information to compressed header (NBO flag, FORMAT flag, MSN)
 - ◆ methods that change field order (STACK encoding methods, U, UNCOMPRESSED)
 - ◆ methods that process more than one field (FORMAT, LIST, CRC, "MACRO").

“MACRO” encoding method

- It is not specified in the EPIC-LITE draft and it was introduced to code multi field rules.
- Multi field rules are BNF rules that process several consecutive header fields that are connected with choice "or" operator.
- Complexity emerges from a fact that if one ORed rule fails, all context, input and output data values have to be restored to a state before ORed rule was applied.
- It is similar to FORMAT method, but choice is indicated through indicator flags, rather than through explicit indication.

Profile data structures

- Profile parsed to linear linked list of dat structures.
- Each structure actually represents one encoding method.

```
typedef struct encoding_method
{
    bool bToolbox;
    VALUE_LIST* value;
    LENGTH* length;
    int offset;
    u16 prob;
    u8 method_id;
    u8 list_order;

    encoding_method* Child;
    encoding_method* Or;
    encoding_method* Next;
    encoding_method* Prev;
};
```

Linear profile

TCP-IP-CO = INFERRED-IP-CHECKSUM(IPv4-co-
header)

TCP-co-header

CRC(8,100%)

IPv4-co-header =

version

header-len

tos-co

ecn

length

ip-id-co

rf-flag-co

df-flag-co

...

version = VALUE(4,4,100%)

header-len = **VALUE(4,5,100%)**

tos-co = **STATIC(99%) | IRREGULAR(6,1%)**

ecn = **IRREGULAR(2,100%)**

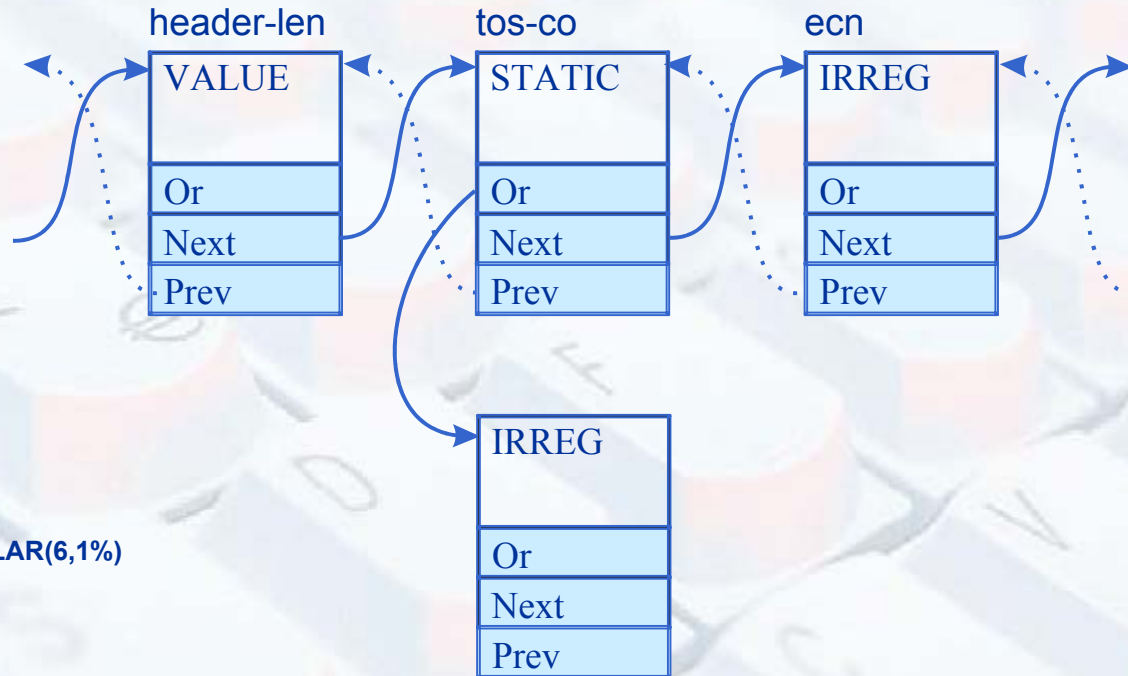
length = INFERRED-SIZE(16,-16)

ip-id-co = NBO(16) ; check byte-order

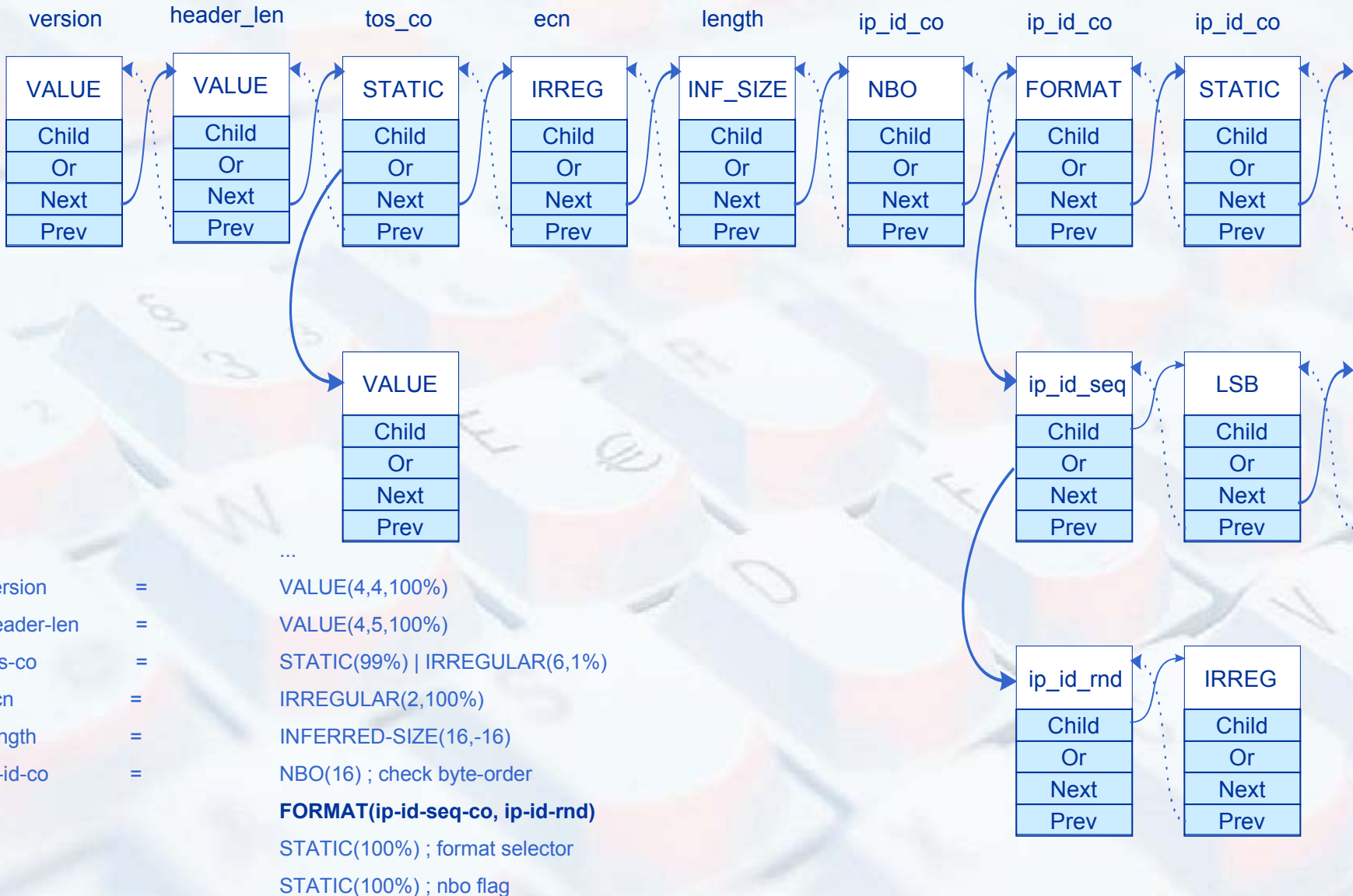
FORMAT(ip-id-seq-co, ip-id-rnd)

STATIC(100%) ; format selector

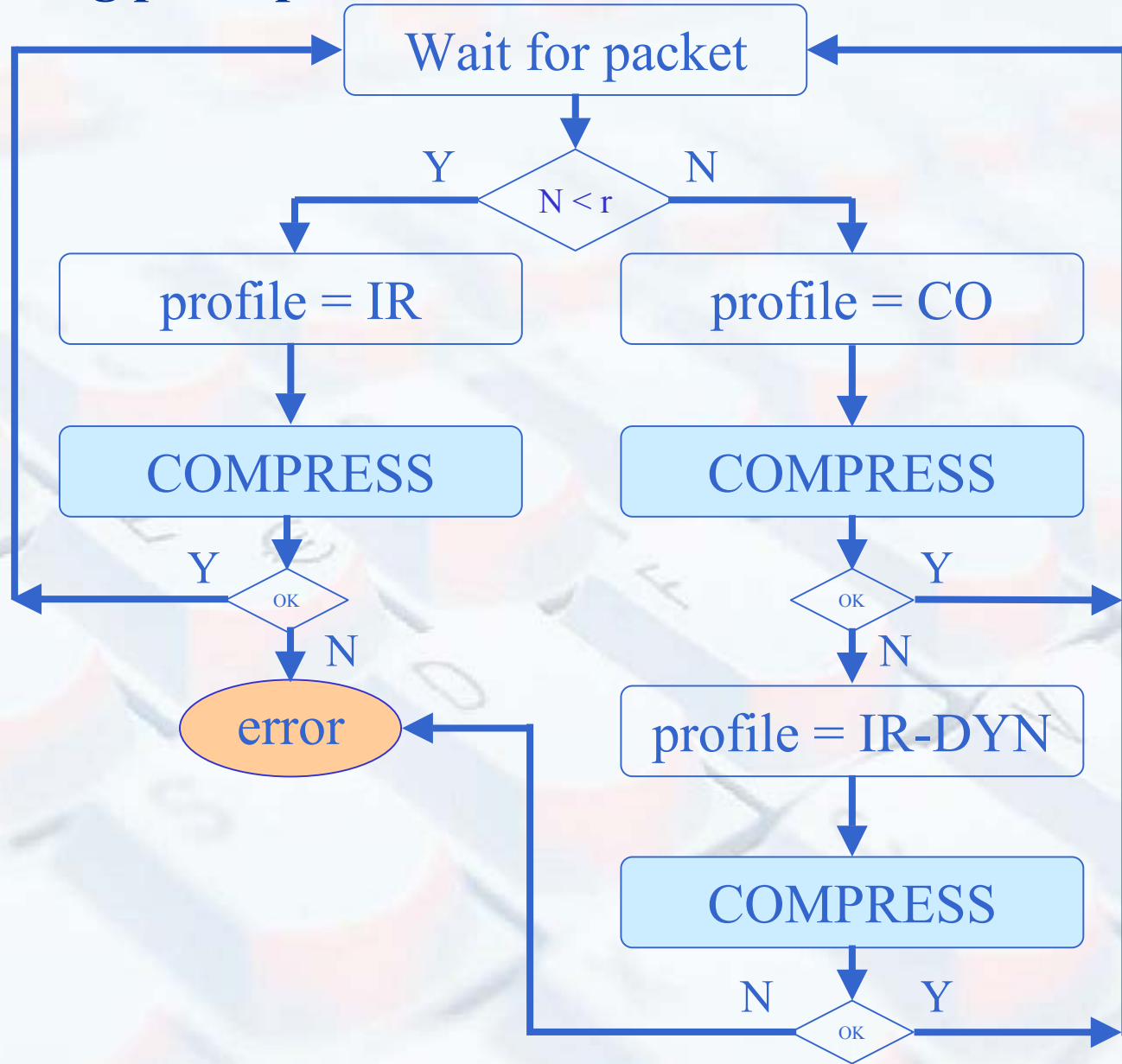
STATIC(100%) ; nbo flag



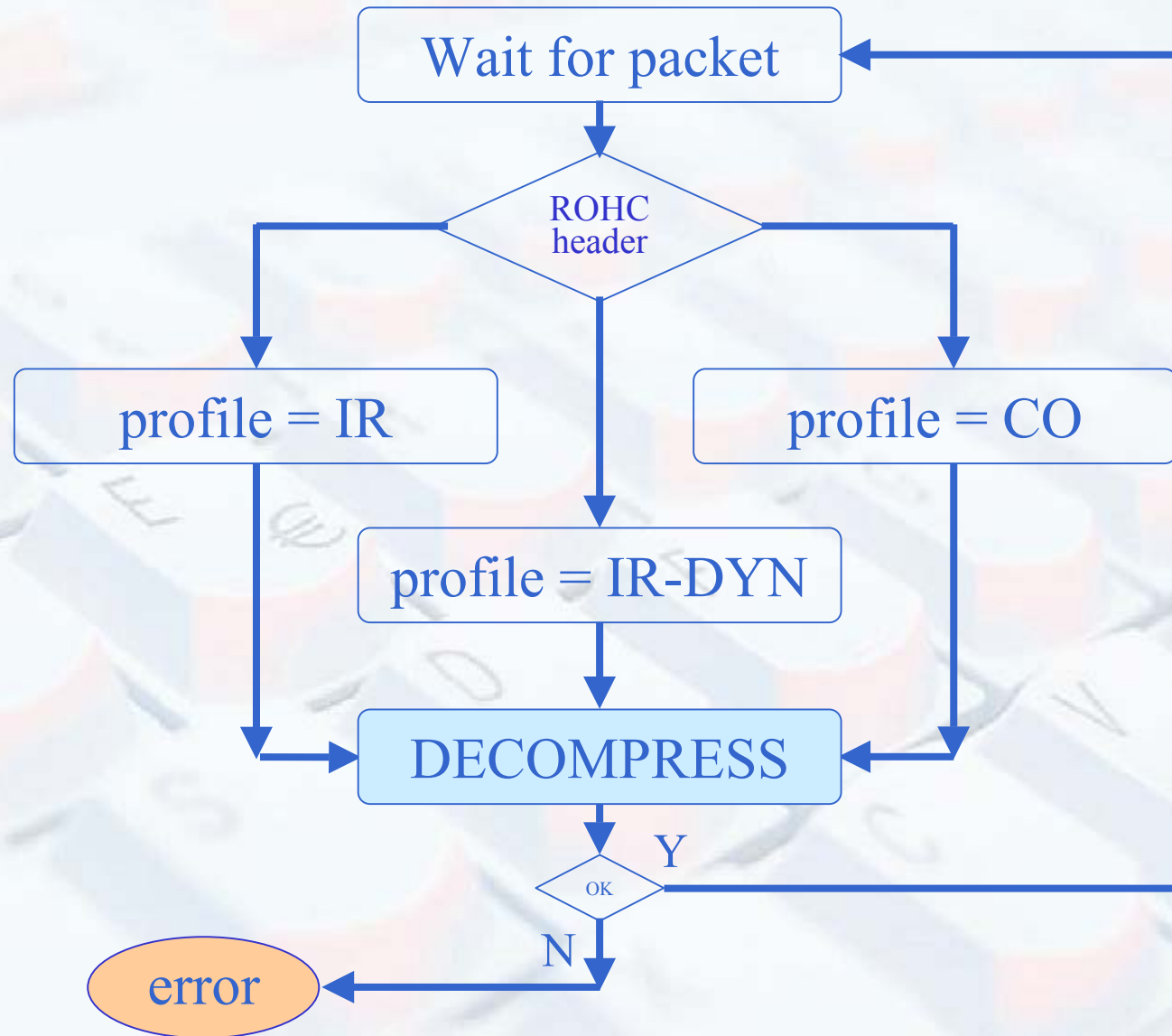
Extended linear profile



Compressor working principles



Decompressor working principles



Conclusion

- Functional compressor and decompressor application can be made based on EPIC-LITE specification.
- Developed implementation includes solution and optimisation of bit manipulation, encoding method classification, and state machine functionality.
- Need for additional method to support ORed rules was recognised, and MACRO method was introduced.
- Extended linear profile structure is successfully applied.
- Future work will provide additional testing of inner parser and extended linear profile with real packet flow.