Encoding of frame information contents.

David Bagby
Bob O'Hara
Dave Roberts
AMD
One AMD Place
Sunnyvale, CA 94088-3453

Rick White
Mark Demange
Motorola
50 East Commerce Dr.
Schaumburg, IL 60173

Jon Rosdahl
Novell
122 East 1700 south
m/s C-22-1
Provo, UT 84606-6194

Paper scope

- Talks only about encoding, not what info is needed in a frame.
- Only addresses "payload" of frame, header encoding is constant.
- Examples assume 94/214a as frame contents.
B2 draft approach

• Elements
  - Singly linked list
  - Element Format
    » Element code, one octet
    » Link field, one octet
      • 1 bit = more elements
      • 7 bits = length of element
    » Element value, 1-n octets
  - EP bit in header indicates if elements are present in frame body.

B2 draft approach

• B2 element encoding problems:
  - Allows arbitrary order of elements
    » Don't want to have to scan entire msg to determine if all needed info is present.
    » Negative added value from allowing arbitrary order.
  - Efficiency
    » Linked list inefficient for fixed length items
      • 1 octet value requires 3 octets of bandwidth
    » Length field good form for variable length items
      • Size limited to 127 octets
      • MSB used as "more elements" bit.
      • To short for some variable length items required
        - Security algorithm dependent information.
    » Boolean elements require 3 octets to represent.
  - Redundant information contained in encoding scheme
    » EP bit
    » Element Code
Proposed improved encoding

- The contents of each frame type are invariant.
  - Uniquely specified by header "type" fields.
  - One purpose of a standard is to create interoperability.
  - Fields within Frames are standardized, they do not "come and go as they please".
  - Therefore the EP bit in the header is redundant and unnecessary - it is removed.

- Disallow arbitrary ordering of fields.
  - Standardize order of fields in frames.
  - When possible, place fixed length fields first.
    » Fixed length fields are octet multiples in size and use only the bandwidth required by the information contained.
  - Place variable length fields after fixed length fields.
    » All variable length fields become
      » All lengths in octet multiples.
      » Use length field appropriate for maximum length of information.

Proposed improved encoding

- Element code and link fields are not needed for fixed length information fields.
  - Elimination improves bandwidth utilization
    » Single octet value goes from 3 octets -> one octet storage

- "Element code" not needed for variable length fields
  - Presence of fields determined by frame type.

- Length field needed for variable length fields
  - Old link field to small.
    » 127 octets
  - Increase to 2 octet length field.
    » some variable length items are defined to have a maximum length
    » in those cases, use an appropriate length field size.
Proposed improved encoding

- Boolean indicators should be either a bit, or a specific value as appropriate.
  - Multiple Boolean bits should be packed into octets when appropriate.

Example detail assuming 94/214a information fields by frame.
**CRC**

- All frames end with CRC
  - Not shown in examples for brevity

**Data frames**

- Subtype = Asynchronous Data
- Subtype = CF Up
- Subtype = CF Down
- All three the same:

  Full Header: 24 octets
  MPDU: \( \leq \) MSDU size
  (frag size dependent)
RTS frame

RTS Header: 16 octets

CTS frame

CTS/ACK Header: 10 octets
ACK frame

CTS/ACK Header: 10 octets

CF-ACK frame

CTS/ACK Header: 10 octets
**POLL frame**

Full Header: 24 octets  
SID: 2 octets

---

**Beacon frame**

Full Header: 24 octets  
Time stamp: 4 octets  
Weight: 2 octets  
Beacon interval: 1 octet  
DTIM period: 1 octet  
DTIM count: 1 octet  
Channel sync: $2 \rightarrow 2 + 2^{16}$ octets  
ESS ID: $1 \rightarrow 1 + 128$ octets  
TIM: $(1 \rightarrow 8$ octets) * $(n - 1) + m$  
Bcast indicator: bit 0 of 1 octet

- $0 =$ no broadcast indication  
- $1 =$ bcast will follow next DTIM
ATIM frame

Full Header: 24 octets

Probe frames

Full Header: 24 octets
Req/Resp: bit 0 of 1 octet
0 = request
1 = response

- Subsequent octets dependent on req/resp Boolean...
Probe frames

- Request:
  - no additional octets.

- Response:
  Time stamp: 4 octets
  Weight: 2 octets
  Beacon interval: 1 octet
  DTIM period: 1 octet
  DTIM count: 1 octet
  Channel sync: $2 \rightarrow 2+2^{16}$ octets
  ESS ID: $1 \rightarrow 1+128$ octets

Association frames

Full Header: 24 octets
Req/Resp: bit 0 of 1 octet
  0 = request
  1 = response

- Request:
  Privacy Alg: 2 octets

- Response:
  Status value: 1 octet
    0 = successful
    $>0$ = error code
  If status value = successful
  SID: 2 octets
Reassociation frames

Full Header: 24 octets

- Req/Resp: bit 0 of 1 octet
  0 = request
  1 = response

- Request:
  Curr AP addr: 6 octets
  Privacy Alg: 2 octets

- Response:
  Status value: 1 octet
    0 = successful
    >0 = error code
  If status value = successful
  SID: 2 octets

Disassociation frame

Full Header: 24 octets

Presentation

Page 11
Privacy frames

- Transaction sequence = 1:

  Full Header: 24 octets
  Trans seq #: 1 octet
  1 = first msg in sequence
  # Algs: 1 octet
  Alg # list: # Algs * 2 octets

Privacy frames

- Transaction sequence = 2:

  Full Header: 24 octets
  Trans seq #: 1 octet
  2 = second msg in sequence
  Status value: 1 octet
  0 = successful
  >0 = error code
  If status value = successful
  Privacy alg #: 2 octets
Authentication frames

• Transaction sequence = 1:

Full Header: 24 octets
Trans seq #: 1 octet
 1 = first msg in auth sequence
# Algs: 1 octet
Alg list: # Algs * 2 octets

Authentication frames

• Transaction sequence = 2:

Full Header: 24 octets
Trans seq #: 1 octet
 2 = second msg in auth sequence
Status value: 1 octet
 0 = successful
 >0 = error code
If status value = successful
Auth alg #: 2 octets
Authentication frames

• Transaction sequence = 3:

Full Header: 24 octets
Trans seq #: 1 octet
3 = third msg in auth sequence
Challenge(s1,s2): 2 -> 2^16 octets

• Transaction sequence = 4:

Full Header: 24 octets
Trans seq #: 1 octet
4 = fourth msg in auth sequence
Ch. Resp (S2, S1): 2 -> 2^16 octet
Challenge (S2, S1): 2 -> 2^16 octets
Authentication contents

- Transaction sequence = 5:

  Full Header: 24 octets
  Trans seq #: 1 octet

  5 = fifth msg in auth sequence

  Ch_Result (S1, S2): 2 -> 2+2^16 octets
  Ch_Response (S1, S2): 2 -> 2+2^16 octets

Authentication contents

- Transaction sequence = 6:

  Full Header: 24 octets
  Trans seq #: 1 octet

  6 = sixth msg in auth sequence

  Ch_Result(S2, S1): 2 -> 2+2^16 octets
Motion:

- That the proposed encodings of frame contents as described in in 94/215 be adopted and that the draft be updated to reflect this.