Network Coding (NC)

CITHN2002 – Summer 2024

Prof. Dr.-Ing. Stephan Günther

Chair of Distributed Systems and Security
School of Computation, Information and Technology
Technical University of Munich

Chapter 1: IEEE 802.11

Chapter 1: IEEE 802.11

IEEE 802.11 frame format

IEEE 802.11 uses three different frametypes:

- Data frames
  - Contain data of any kind (both user data and "management traffic" such as ARP, neighbor discovery, DNS, etc.)
  - Payload may be encrypted
  - Various subtypes (e. g. QoS and many special formats for networks with AP)

IEEE 802.11 frame format

IEEE 802.11 uses three different frametypes:

- Data frames
  - Contain data of any kind (both user data and "management traffic" such as ARP, neighbor discovery, DNS, etc.)
  - Payload may be encrypted
  - Various subtypes (e. g. QoS and many special formats for networks with AP)
- Management frames
  - Management traffic between stations, in particular to associate to an AP
  - No encryption
  - Various subtypes (e. g. beacons, association requests, etc.)

IEEE 802.11 frame format

IEEE 802.11 uses three different frametypes:

- Data frames
  - Contain data of any kind (both user data and "management traffic" such as ARP, neighbor discovery, DNS, etc.)
  - Payload may be encrypted
  - Various subtypes (e. g. QoS and many special formats for networks with AP)
- Management frames
  - Management traffic between stations, in particular to associate to an AP
  - No encryption
  - Various subtypes (e. g. beacons, association requests, etc.)
- Control frames
  - Frames assisting in media access
  - No encryption
  - Various subtypes (e. g. RTS / CTS, ACK, etc.)

Each frame type (even subtypes) has custom headers
⇒ variable length headers *without* explicit length specification

IEEE 802.11 frame format

The generic frame format looks as follows:

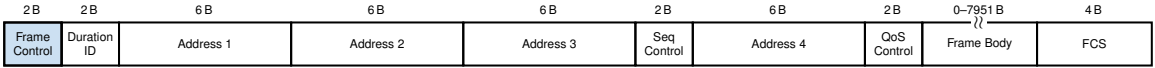


Figure 1: IEEE 802.11 generic header [1]

Frame control

- Defines frame type and subtype
  - Different frame types may lack some of the header fields
  - Variable header lengths start here
- Controls how MAC addresses shall be interpreted
- Fragmentation control
- Indicates whether or not the payload is encrypted (but not how it is encrypted)
- etc.

IEEE 802.11 frame format

The generic frame format looks as follows:

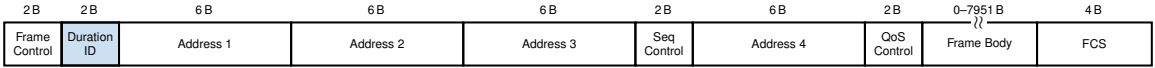


Figure 1: IEEE 802.11 generic header [1]

Duration / ID

- Meaning and content differs between frame types
- One application is to assist in virtual carrier sensing, i. e., the expected duration of a transmission is specified
  - Allows other stations to return to low-power state
  - Potential for abuse?

IEEE 802.11 frame format

The generic frame format looks as follows:

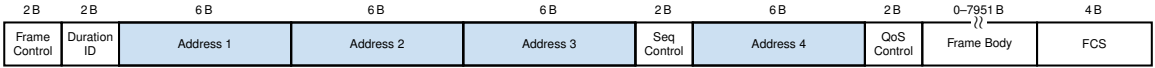


Figure 1: IEEE 802.11 generic header [1]

4 MAC addresses

- Interpretation depends on the ToDS / FromDS bits in the frame control field
- Not all addresses may be present
  - Data frames in infrastructure mode commonly use 3 addresses
  - Data frames in ad-hoc mode use 2 addresses
  - Acknowledgements use a single address only (which one?)
- MAC addresses are the same as with IEEE 802.3

IEEE 802.11 frame format

The generic frame format looks as follows:

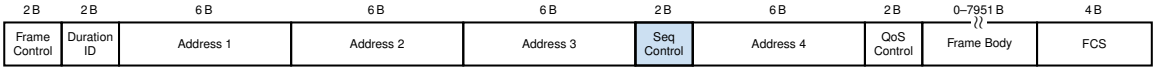


Figure 1: IEEE 802.11 generic header [1]

Sequence Control

- Consists of a fragment number (4 bit) and a sequence number (12 bit)
- Fragment number is used for fragmentation and reassembly of frames
- Sequence number is needed for link-layer acknowledgements

IEEE 802.11 frame format

The generic frame format looks as follows:

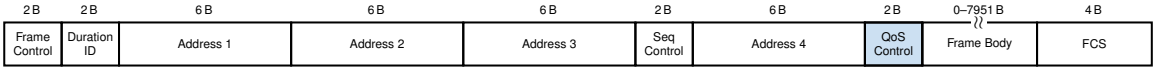


Figure 1: IEEE 802.11 generic header [1]

QoS control

- Used for quality of service (traffic classes, priorities, etc.)
- Only used if QoS is enabled, which implies a different frame type for data frames

IEEE 802.11 frame format

The generic frame format looks as follows:

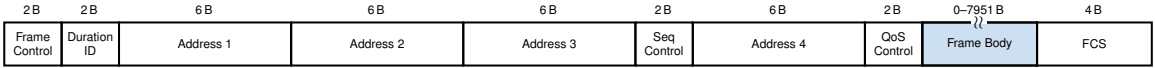


Figure 1: IEEE 802.11 generic header [1]

Frame body

- Everything that is considered as payload
- May be encrypted
- Contains other headers (even before the layer 3 header), e. g.:
  - headers specific to encryption (WEP, WPA)
  - SNAP header (variable length header, function similar to the EtherType field in IEEE 802.3)
- Maximum length depends on the PHY in use

IEEE 802.11 frame format

The generic frame format looks as follows:

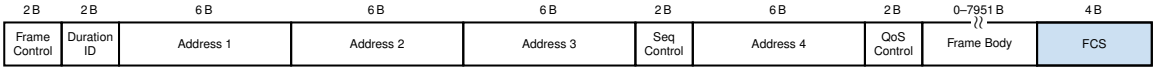


Figure 1: IEEE 802.11 generic header [1]

FCS

- Frame check sequence to detect transmission errors
- 32 bit CRC with specific register initialization / inversion
- Generally calculated by hardware or drivers

IEEE 802.11 frame format

The generic frame format looks as follows:

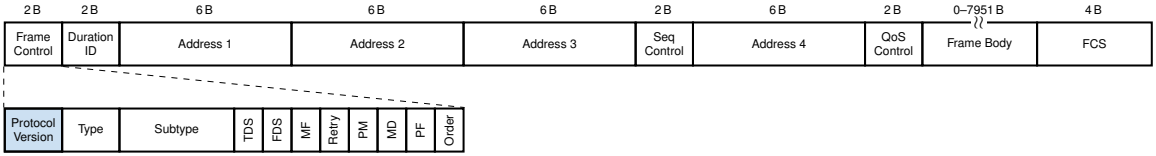


Figure 1: IEEE 802.11 generic header [1]

Protocol Version

- Must be set to 0 on current hardware
- Drivers will most likely drop frames with different version

IEEE 802.11 frame format

The generic frame format looks as follows:

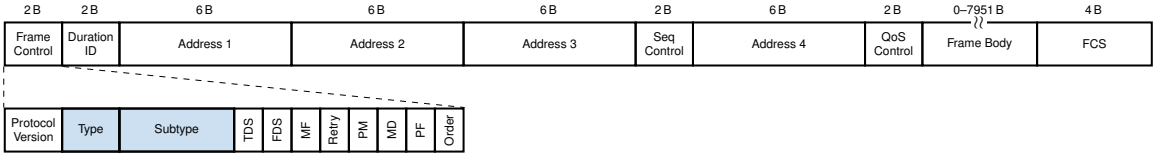


Figure 1: IEEE 802.11 generic header [1]

Type and Subtype

- Defines the type (data, management, or control) and subtype (e. g. QoS data) of frames
- Type and subtype are simply ORed, e. g.
  `IEEE80211_FTYPE_CTL | IEEE80211_STYPE_ACK`

IEEE 802.11 frame format

The generic frame format looks as follows:

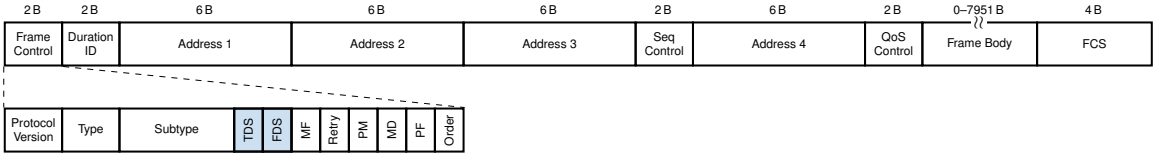


Figure 1: IEEE 802.11 generic header [1]

ToDS and FromDS

- Define how MAC addresses are interpreted:
  - Receiver Address (RA), i. e., the receiving STA (possibly along a path of multiple hops)
  - Transmitter Address (TA), i. e., the transmitting STA
  - Destination Address (DA), i. e., final destination of a frame within the actual L3 broadcast domain
  - Source Adress (SA), i. e., original source of a frame within the actual L3 broadcast domain

ToDS	FromDS	Address 1	Address 2	Address 3	Address 4
0	0	RA = DA	TA = SA	BSSID	n/a
0	1	RA = DA	TA = BSSID	SA	n/a
1	0	RA = BSSID	TA = SA	DA	n/a
1	1	RA	TA	DA	SA

IEEE 802.11 frame format

The generic frame format looks as follows:

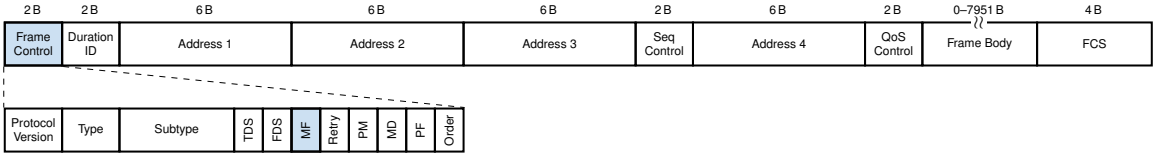


Figure 1: IEEE 802.11 generic header [1]

More Fragments

- Indicates whether or not the frame contains another fragment of the current MSDU
- Used to reassemble the MSDU before forwarding to higher layers
- Set to 0 for all control frames

IEEE 802.11 frame format

The generic frame format looks as follows:

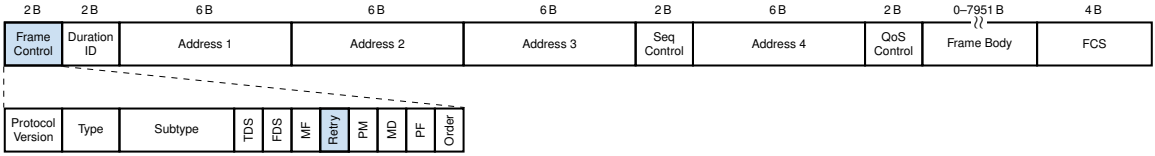


Figure 1: IEEE 802.11 generic header [1]

Retry

- Indicates that the current frame is a retry, i. e., the frame has been sent before but no ACK has been received
- Helps the receiver to eliminate duplicate frames
- Why is it set to zero for all broadcast and multicast frames?

IEEE 802.11 frame format

The generic frame format looks as follows:

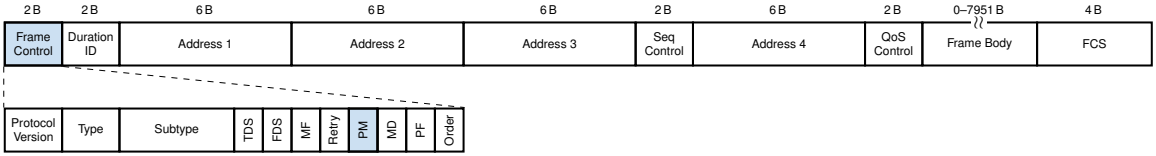


Figure 1: IEEE 802.11 generic header [1]

Power Management

- Indicates the power management mode of the transmitter after successful transmission of the current frame (or sequence of frames)
- Set to 0 (no power save) if transmitter is an AP

IEEE 802.11 frame format

The generic frame format looks as follows:

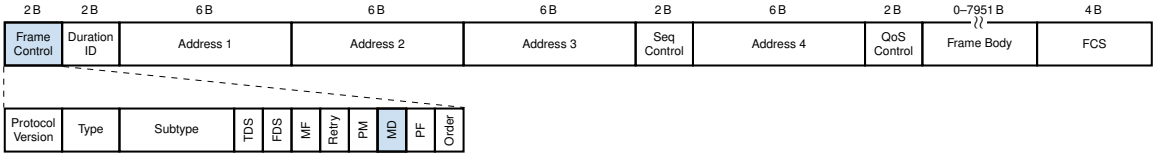


Figure 1: IEEE 802.11 generic header [1]

More Data

- Indicates that the transmitter has more data destined for the same receiver
- Used to indicate to a STA that power save should be suspended

IEEE 802.11 frame format

The generic frame format looks as follows:

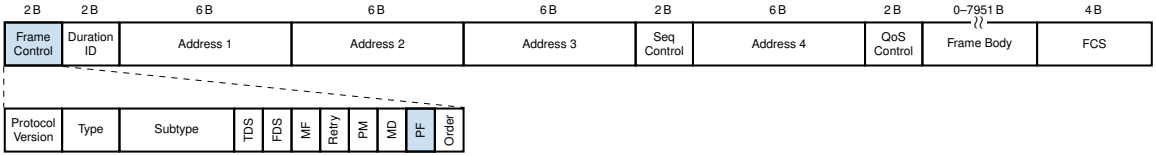


Figure 1: IEEE 802.11 generic header [1]

Protected Frame

- Originally used to indicate WEP encryption
- It is now used to indicate that the frame body contains some information about how the content is protected

IEEE 802.11 frame format

The generic frame format looks as follows:

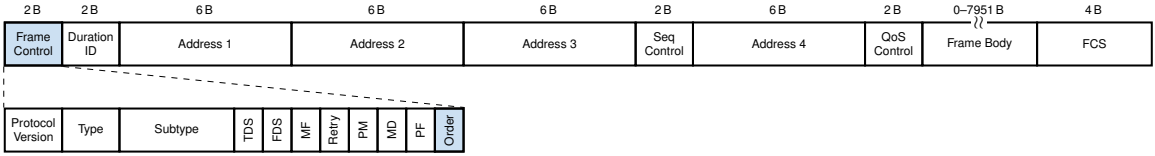


Figure 1: IEEE 802.11 generic header [1]

Order

- Only used for non-QoS data frames that contain an MSDU being transferred using the strictly ordered service class
  - The transmitter may intentionally reorder frames to increase the likelihood of successfull delivery
  - If set, the receiver is responsible to buffer and reorder frames such that data is not sent out-of-order to higher layers
- It is “abused” by QoS STAs to indicate whether or not an additional HT-control field (not shown in Figure 1) is present

IEEE 802.11 frame format

There are (at least) 2 weird things on this format:

1. There is nothing that specifies the next layer protocol
2. The maximum frame body size of 7951 B exceeds the common MTU of 1500 B quite a bit

IEEE 802.11 frame format

There are (at least) 2 weird things on this format:

1. There is nothing that specifies the next layer protocol
2. The maximum frame body size of 7951 B exceeds the common MTU of 1500 B quite a bit

The first one is quickly explained:

- The frame body contains a SNAP header (subnetwork access protocol)
- It specifies the next layer protocol, whatever it might be
- Unfortunately, the SNAP header is again of variable length
- There might be encryption headers before the SNAP header

The second one takes a bit longer, more on that later.

CSMA / CA is used:

- Sense the medium for ongoing transmission before transmitting ("listen before talk")
- Since collisions cannot be reliably detected (hidden stations, sensing while transmitting), collisions have to be avoided

How is collision avoidance implemented in IEEE 802.11:

- So called coordination functions define the collision avoidance scheme
- The most basic method is the distributed coordination function (DCF)
- All other methods are based or derived from the DCF
- Optionally, nodes may use RTS / CTS protection

Distributed coordination function (DCF)

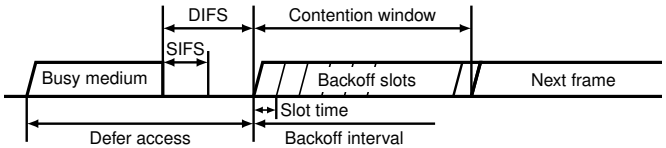


Figure 1: Media access via distributed coordination function (DCF)

Assuming that a node is backlogged:

1. Medium is sensed until it becomes idle
2. The medium has to be idle for a specific minimum idle time (called inter frame spacing)
3. The node draws an independently and uniformly distributed number from a contention window
4. The node further defers transmission for this number of time slots
   - 4.1 After this backoff and if the medium is still idle, the node starts transmitting
   - 4.2 Otherwise transmission is deferred and the process starts from scratch when the medium becomes idle again

- In contrast to IEEE 802.3, the contention window $W = \{0, 1, \ldots, m\}$ has a minimum size of $m > 0$.
- If a transmission error occurs, i. e., a data frame is not acknowledged by the receiver, the contention window is increased:

$$m \equiv C(n) = \min \left\{ 2^{n+k} - 1, 255 \right\},$$

  where k defines the minimum size (depends on the coordination function) and n is the number of failed transmission attempts.
- A common value for $C(0)$ is 15.

How severe is it?

- Let the random variable C_n denote the number of backoff slots drawn for a given transmission attempt.
- Assuming that only one node is backlogged and no transmission errors, there is an additional idle time of $\mathrm{E}[C_0]$.

Example: HT mixed mode, 5 GHz (802.11n)

- Slot time is 9 µs, $C(0) = 15 \Rightarrow 67.5$ µs
- Inter frame spacing with DCF adds another 34 µs
- The average total delay for media access (without PHY headers) is therefore $\Delta t = 101.5$ µs

Example: HT mixed mode, 5 GHz (802.11n)

- Slot time is 9 µs, $C(0) = 15 \Rightarrow 67.5$ µs
- Inter frame spacing with DCF adds another 34 µs
- The average total delay for media access (without PHY headers) is therefore $\Delta t = 101.5$ µs

How much time is needed for the actual tansmission?

- Assume an MPDU[1] of $l = 1500$ B, and forget about any other overhead that might exist
- Assume a bit rate of $r = 150$ Mbit/s (maximum rate of 802.11n with one antenna)
- The actual transmission lasts only $t = l/r = 80$ µs

[1] MAC PDU

Example: HT mixed mode, 5 GHz (802.11n)

- Slot time is 9 µs, $C(0) = 15 \Rightarrow 67.5$ µs
- Inter frame spacing with DCF adds another 34 µs
- The average total delay for media access (without PHY headers) is therefore $\Delta t = 101.5$ µs

How much time is needed for the actual tansmission?

- Assume an MPDU[1] of $l = 1500$ B, and forget about any other overhead that might exist
- Assume a bit rate of $r = 150$ Mbit/s (maximum rate of 802.11n with one antenna)
- The actual transmission lasts only $t = l/r = 80$ µs

$\Rightarrow$ Media access takes longer than the actual transmission!

[1] MAC PDU

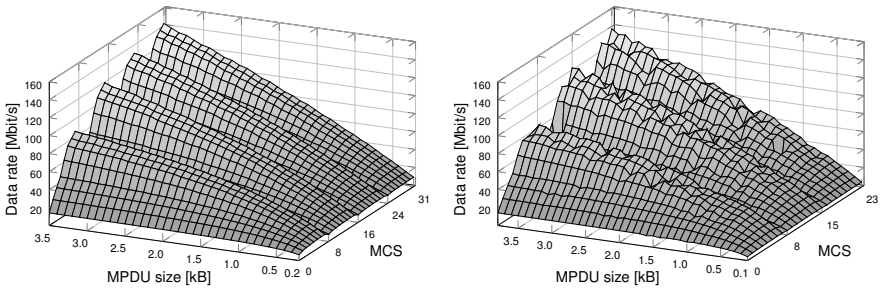


Figure 2: TX simulation (a) and RX measurement (b) using AR9390 chipsets [2]

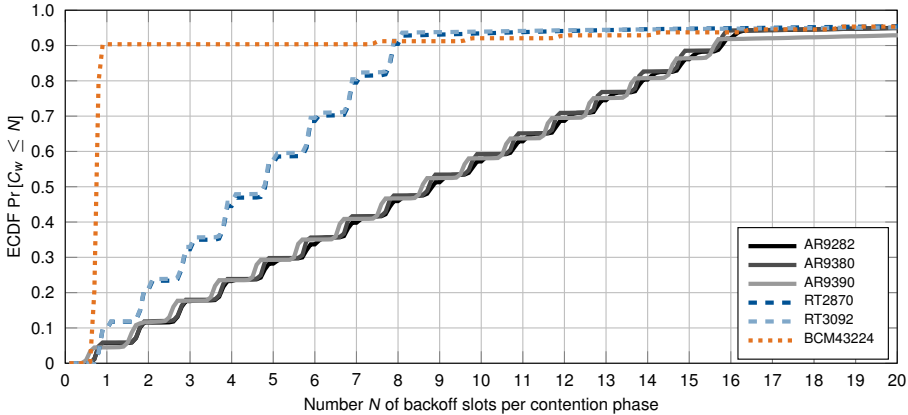
IEEE 802.11 media access

- The initial size of the contention window depends on the standard in use
- A common value is 15 slot times, i. e., the number of slot times to wait before transmission is drawn uniformly and independently distributed from the set $\{0, 1, \ldots, 15\}$

Do devices adhere to that rule? [2]

- The initial size of the contention window depends on the standard in use
- A common value is 15 slot times, i. e., the number of slot times to wait before transmission is drawn uniformly and independently distributed from the set $\{0, 1, \ldots, 15\}$

Do devices adhere to that rule? [2]



What does that mean for MAC fairness? [2]

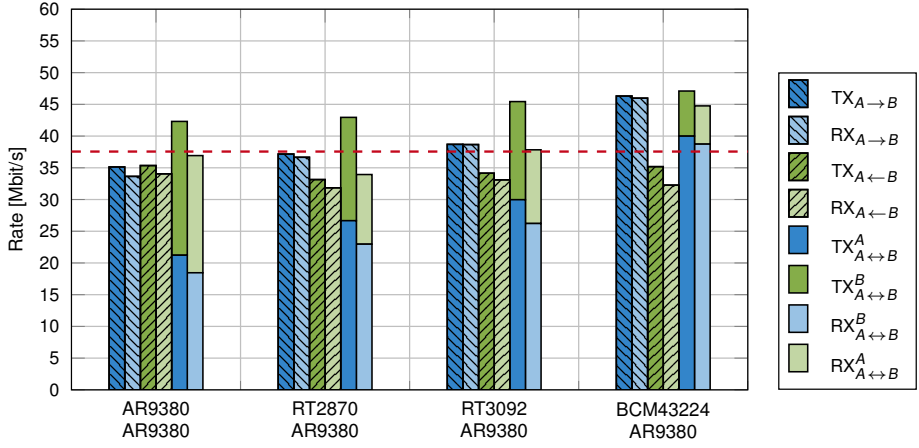


Figure 3: $TX_{A \to B}$ is the rate node A is transmitting. $TX^{A}_{A \leftrightarrow B}$ is the rate at which node A is transmitting when both A and B are contending for media access. RX are the respective goodputs, i. e., the rate at which nodes are receiving under the respective conditions. The dashed line represents the upper achievable upper bound when adhering to the default contention window size.

IEEE 802.11 media access

Besides large MPDUs, IEEE 802.11 has several mechanisms to reduce MAC delays:

- APs help to coordinate medium access, e. g. point coordination function (PCF) etc.
- Stations may aggregate multiple frames into an AMPDU, which are sent without repeated media access
- etc.

Most mechanisms require an AP, i. e., infrastructure mode, and more complex coordination functions.

IEEE 802.11 service sets

- Basic service set (BSS) or infrastructure mode consists of an AP with all associated STAs
  - Identified by its BSSID (usually the MAC address of the AP)
  - STAs do not communicate directly with each other, the AP relays messages

IEEE 802.11 service sets

- Basic service set (BSS) or infrastructure mode consists of an AP with all associated STAs
  - Identified by its BSSID (usually the MAC address of the AP)
  - STAs do not communicate directly with each other, the AP relays messages
- Extended service set (ESS) or distribution system (DS) is a set of connected APs (e. g. over Ethernet) and their associated STAs
  - Identified by its ESSID (that is what you see when searching for networks)
  - APs relay messages to other APs

IEEE 802.11 service sets

- Basic service set (BSS) or infrastructure mode consists of an AP with all associated STAs
  - Identified by its BSSID (usually the MAC address of the AP)
  - STAs do not communicate directly with each other, the AP relays messages
- Extended service set (ESS) or distribution system (DS) is a set of connected APs (e. g. over Ethernet) and their associated STAs
  - Identified by its ESSID (that is what you see when searching for networks)
  - APs relay messages to other APs
- Independent basic service set (IBSS) or ad-hoc mode is a set of STAs communicating directly with each other without AP
  - STAs can communicate only with other STAs in range
  - STAs do not automatically relay messages on behalf of others
  - An IBSS may form a mesh network when suitable routing protocols are installed

How is a BSS formed?

- The AP broadcasts beacons in regular intervals, which contain
  - the BSSID (and ESSID),
  - channel, frequency, supported hardware modes, data rates,
  - and many more information.
- When an STA *joins* a BSS, a four-way-handshake is performed.
- Afterwards, the STA is associated, i. e., link-layer connectivity is established.

After association many more things might happen, e. g.

- negotiation of encryption, authentication etc.,
- obtaining a network layer address,
- ...

Radiotap

The Linux kernel allows to pass per-frame options to / from device drivers:

- A so-called radiotap header is prepended to each frame.
- The radiotap header is **not** transmitted but
  - pass options to the driver, e. g. rate at which a packet should be sent, and
  - retrieve information about both sent and received frames.
- Under normal operation, the radiotap header is never visible: it is added / stripped by the `mac80211` subsystem of the kernel.
- If an interface is set to monitor mode, radiotap headers are expected from and passed to userspace applications.

Radiotap

The Linux kernel allows to pass per-frame options to / from device drivers:

- A so-called radiotap header is prepended to each frame.
- The radiotap header is **not** transmitted but
  - pass options to the driver, e. g. rate at which a packet should be sent, and
  - retrieve information about both sent and received frames.
- Under normal operation, the radiotap header is never visible: it is added / stripped by the `mac80211` subsystem of the kernel.
- If an interface is set to monitor mode, radiotap headers are expected from and passed to userspace applications.

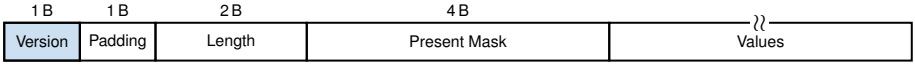


Figure 4: Radiotap header

- Version of the radiotap header
- Currently always set to zero

Radiotap

The Linux kernel allows to pass per-frame options to / from device drivers:

- A so-called radiotap header is prepended to each frame.
- The radiotap header is **not** transmitted but
  - pass options to the driver, e. g. rate at which a packet should be sent, and
  - retrieve information about both sent and received frames.
- Under normal operation, the radiotap header is never visible: it is added / stripped by the `mac80211` subsystem of the kernel.
- If an interface is set to monitor mode, radiotap headers are expected from and passed to userspace applications.

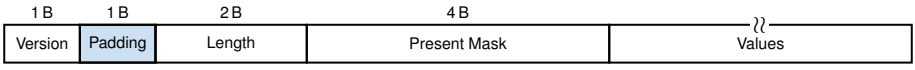


Figure 4: Radiotap header

- Currently unused
- Aligns the length field to a word boundary (word = 2 B)

Radiotap

The Linux kernel allows to pass per-frame options to / from device drivers:

- A so-called radiotap header is prepended to each frame.
- The radiotap header is **not** transmitted but
  - pass options to the driver, e. g. rate at which a packet should be sent, and
  - retrieve information about both sent and received frames.
- Under normal operation, the radiotap header is never visible: it is added / stripped by the `mac80211` subsystem of the kernel.
- If an interface is set to monitor mode, radiotap headers are expected from and passed to userspace applications.

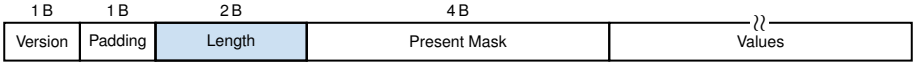


Figure 4: Radiotap header

- Indicates the length of the radiotap header (header plus any options)
- Length is indicated in multiples of 1 B
- All fields in the Radiotap header are little-endian.

Radiotap

The Linux kernel allows to pass per-frame options to / from device drivers:

- A so-called radiotap header is prepended to each frame.
- The radiotap header is **not** transmitted but
  - pass options to the driver, e. g. rate at which a packet should be sent, and
  - retrieve information about both sent and received frames.
- Under normal operation, the radiotap header is never visible: it is added / stripped by the `mac80211` subsystem of the kernel.
- If an interface is set to monitor mode, radiotap headers are expected from and passed to userspace applications.

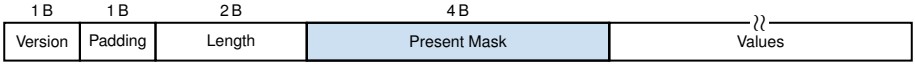


Figure 4: Radiotap header

- Bit mask indicating which options (values) are present in the variable length section of the header
- The highest order bit indicates whether or not there are additional present masks (!)

Radiotap

The Linux kernel allows to pass per-frame options to / from device drivers:

- A so-called radiotap header is prepended to each frame.
- The radiotap header is **not** transmitted but
  - pass options to the driver, e. g. rate at which a packet should be sent, and
  - retrieve information about both sent and received frames.
- Under normal operation, the radiotap header is never visible: it is added / stripped by the `mac80211` subsystem of the kernel.
- If an interface is set to monitor mode, radiotap headers are expected from and passed to userspace applications.

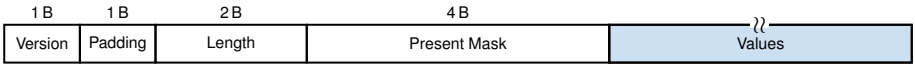


Figure 4: Radiotap header

- Variable length field containing radiotap options
- The length of each option is fixed, but different options may have different sizes
- Options must be naturally aligned, e. g. a 2 B option must be aligned to multiples of 2 B in memory
- To achieve alignment, padding is inserted when needed

Radiotap header present mask

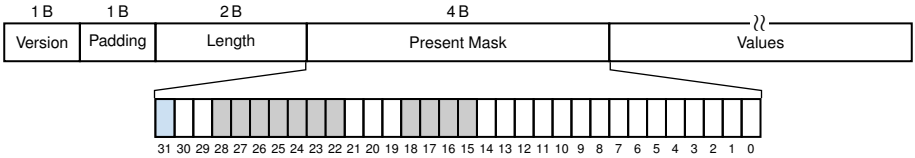


Figure 5: Radiotap header present mask

Bit 31: EXT

- If set to 1, another present mask follows
- If set to 0, options (values) follow to the current mask

Radiotap header present mask

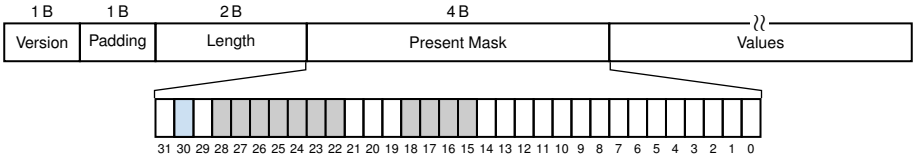


Figure 5: Radiotap header present mask

Bit 30: VENDOR_NAMESPACE

Radiotap header present mask

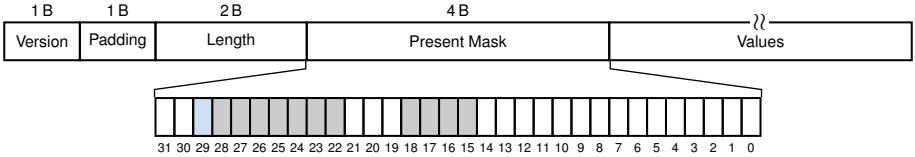


Figure 5: Radiotap header present mask

Bit 29: RADIOTAP_NAMESPACE

Radiotap header present mask

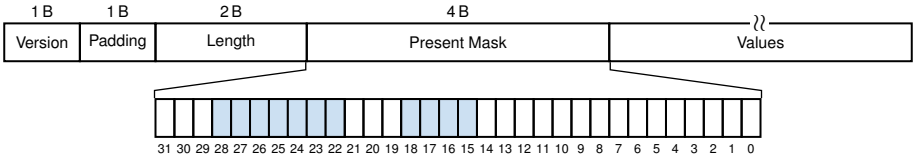


Figure 5: Radiotap header present mask

Bit 28 – 22 and 18 – 15: *inofficial*

- Not defined by the standard
- Linux uses them anyways

Radiotap header present mask

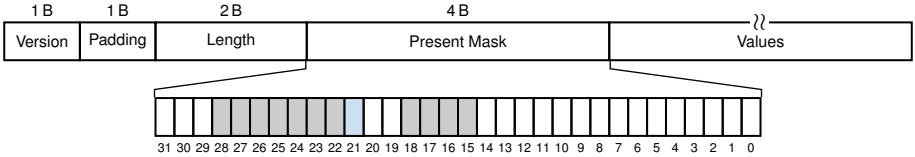


Figure 5: Radiotap header present mask

Bit 21: VHT

Specifies the modulation and coding scheme (MCS) and other parameters for VHT transmissions (IEEE 802.11ac):

- 2 B `known_information` bit mask:
- 1 B flags:
  - guard interval (spectral distance between subchannels)
  - ...
- 1 B bandwidth
- 4 · 1 B MCS index and number of spatial streams
- ...

Radiotap header present mask

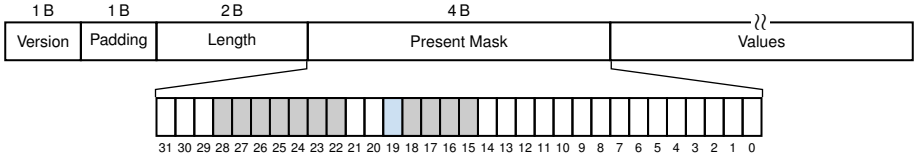


Figure 5: Radiotap header present mask

Bit 19: MCS

Specifies the MCS and other parameters for HT transmissions (IEEE 802.11n):

- MCS index (typically 0 – 31)
  - 0 – 7 ⇒ 1 spatial stream
  - 8 – 15 ⇒ 2 spatial stream
  - 16 – 23 ⇒ 3 spatial stream
  - 24 – 31 ⇒ 4 spatial stream
- 1 B `known_information` bit mask:
  - 1 B flags:
    - guard interval (spectral distance between subchannels)
    - ...
- 1 B MCS index

Radiotap header present mask

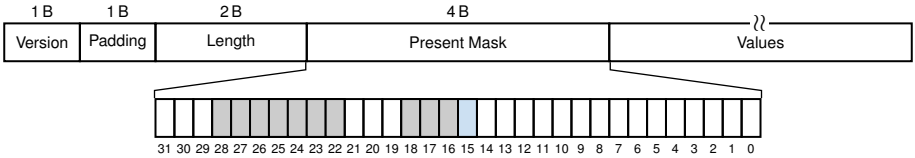


Figure 5: Radiotap header present mask

Bit 15: TX_FLAGS (inofficial)

Controls various media access settings: (IEEE 802.11n)

- Whether or not link-layer acknowledgements are expected for unicast transmissions
- RTS / CTS protection
- CTS-to-self
- ...

Radiotap header present mask

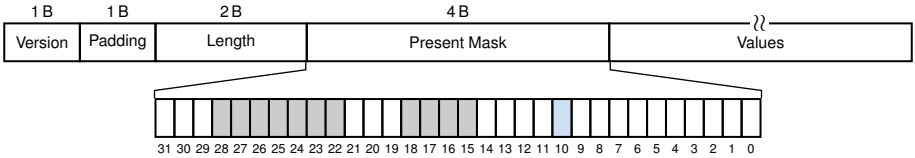


Figure 5: Radiotap header present mask

Bit 10: DBM_TX_POWER

Power at which packets are transmitted (normalized to 1 mW).

Radiotap header present mask

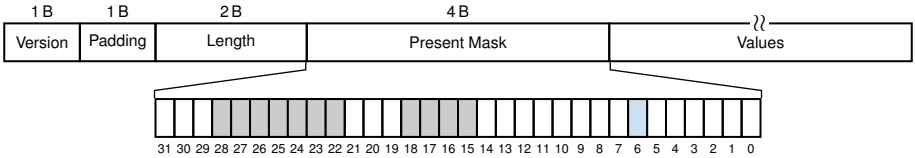


Figure 5: Radiotap header present mask

Bit 6: DBM_ANTNOISE

Noise power during reception of a frame (normalized to 1 mW).

Radiotap header present mask

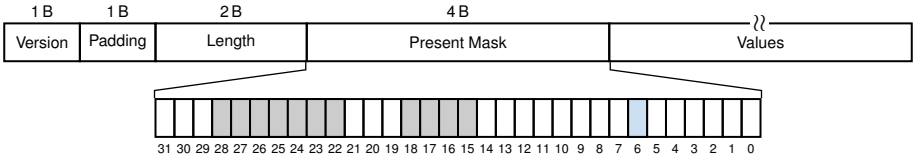


Figure 5: Radiotap header present mask

Bit 6: DBM_ANTNOISE

Noise power during reception of a frame (normalized to 1 mW).

We can compute the maximum achievable data rate using the Shannon-Hartley theorem:

$$
\begin{aligned}
r_{\max} &= B \log_2 \left(1 + \frac{P_S}{P_N}\right) \\
&= B \log_2 \left(1 + 10^{\mathrm{SNR_{dB}}/10}\right) \\
&= B \log_2 \left(1 + 10^{(\text{antenna signal} - \text{antenna noise})/10}\right)
\end{aligned}
$$

Radiotap header present mask

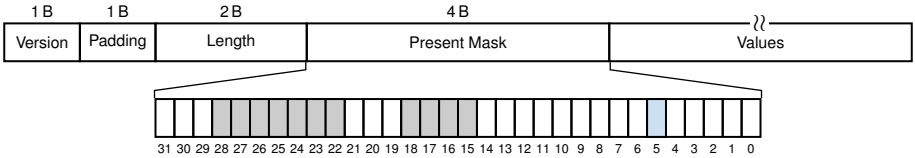


Figure 5: Radiotap header present mask

Bit 5: DBM_ANTSIGNA

Received signal power in dBm (dB difference from 1 mW).

Radiotap header present mask

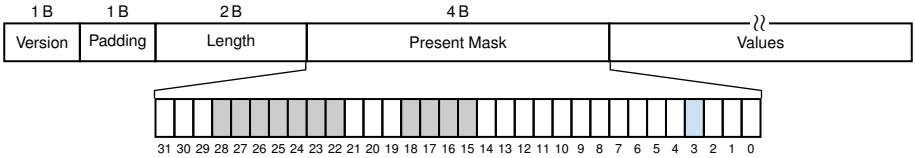


Figure 5: Radiotap header present mask

Bit 3: CHANNEL

Channel frequency in MHz and various flags

- CCK (complementary code keying)
- OFDM (orthogonal frequency division multiplex)
- 2.4 GHz / 5 GHz
- ...

Radiotap header present mask

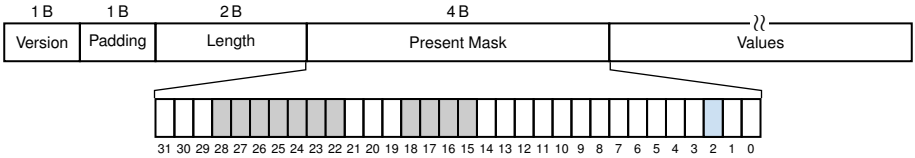


Figure 5: Radiotap header present mask

Bit 2: RATE

TX / RX rate in multiples of 500 kbit/s (IEEE 802.11a/b/g only).

Radiotap header present mask

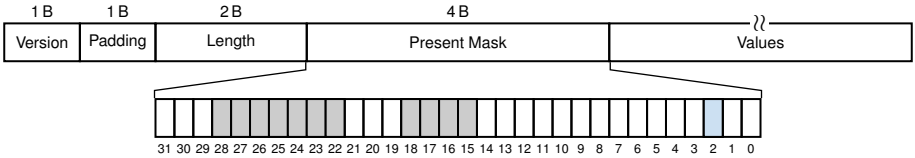


Figure 5: Radiotap header present mask

Bit 2: RATE

TX / RX rate in multiples of 500 kbit/s (IEEE 802.11a/b/g only).

Why 500 kbit/s? (lowest possible rate is 1 Mbit/s)

Radiotap header present mask

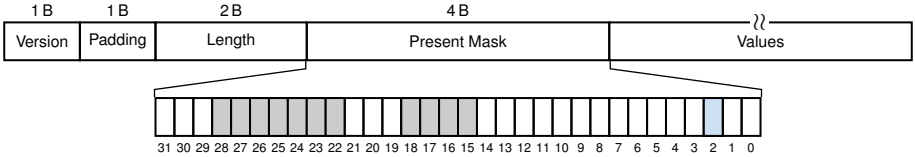


Figure 5: Radiotap header present mask

Bit 2: RATE

TX / RX rate in multiples of 500 kbit/s (IEEE 802.11a/b/g only).

Why 500 kbit/s? (lowest possible rate is 1 Mbit/s)

Standard	Band	Modulation	Rates [Mbit/s]
IEEE 802.11	2.4 GHz	DSSS	1, 2
IEEE 802.11a/g	5 / 2.4 GHz	OFDM	6, 9, 12, 18, 24, 36, 48, 54
IEEE 802.11b	2.4 GHz	CCK	5.5, 11

Radiotap header present mask

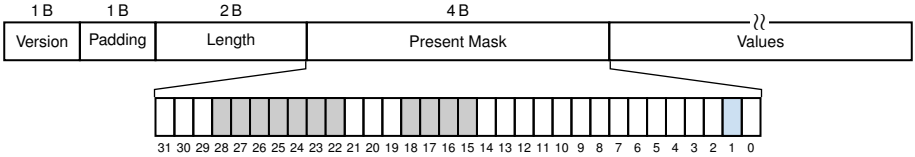


Figure 5: Radiotap header present mask

Bit 1: FLAGS

Various properties of a received frame

- short preamble
- encrypted
- fragmentation
- includes FCS (frame check sequence)
- failed FCS
- short guard interval
- ...

Radiotap header present mask

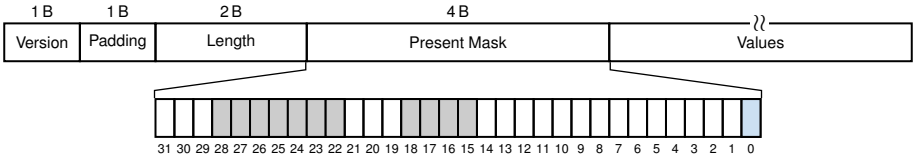


Figure 5: Radiotap header present mask

Bit 0: TSFT

Timestamp for recevied frames (requires support by drivers / hardware)

- measured in µs
- no absolute time

Radiotap values

- Values are appended in order (bit number in present mask)
- Values are aligned on their respective field size
- Length is implicit
- All values are little-endian

Radiotap values

- Values are appended in order (bit number in present mask)
- Values are aligned on their respective field size
- Length is implicit
- All values are little-endian

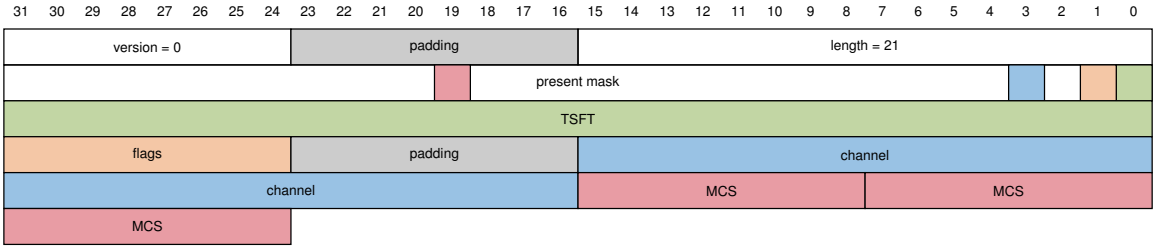


Figure 6: Radiotap example

Possible project: Frame injection & rate control on IEEE 802.11ac devices

- Never worked
- Kernel is in general prepared for it
- Lack of driver / firmware supported
- Investigate the current state of drivers

Bibliography

Bibliography

[1] I. . S. W. Group.
Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification.
IEEE Std 802.11-2012, IEEE, 2012.

[2] S. M. Günther, M. Leclaire, J. Michaelis, and G. Carle.
Analysis of Injection Capabilities and Media Access of ieee 802.11 Hardware in Monitor Mode.
In *IEEE Symposium on Network Operations and Management (NOMS)*, Kraków, Poland, 5 2014.