OpenBSD meets 802.11n

Stefan Sperling <stsp@openbsd.org>

EuroBSDcon 2016

Introduction to 802.11n

The 802.11n standard improves upon earlier standards:

- full backwards compatibility to 802.11a/b/g
- supports data rates up to 600 Mbit/s[1]
- improved signal reception at edges of WLAN cell

Standard was ratified in 2012. Supported by virtually all wifi devices sold since.

[1] maximum data rate of 11a/g standards was 54Mbit/s

Features introduced in 802.11n (overview)

Major technological advancements:

- MIMO (Multiple Input Multiple Output) radio transmission
- channel width can be increased from 20MHz to 40MHz
- frame aggregation

802.11n devices use a High Throughput (HT) PHY. Several improvements relative to earlier PHYs:

- More OFDM subcarriers
- Enhanced FEC code rate
- Short Guard Interval (SGI)
- (there are some more which we won't discuss in this talk)

Additional OFDM subcarriers

802.11a/g divide 20MHz wide channel into 53 subcarriers, each 0.3125MHz wide.

- 4 subcarriers transmit a reference phase
- The channel's centre subcarrier transmits zeros
- Remaining 48 subcarriers transmit data
- Some channel space is left unused: $20/0.3125 = 64$

802.11n adds 4 data subcarriers filling unused space. Maximum data rate goes up from 54Mbit/s to 58.5Mbit/s.

FEC Code Rate

802.11 uses forward error correction (FEC).
Lower data rates transmit many bits redundantly:

- Transmitter adds redundant bits to transmitted data stream
- Receiver applies FEC to detect and correct bit errors

Higher data rates achieved in part by “stealing” redundant bits:

- Transmitter omits some bits which FEC can calculate
- Receiver inserts dummy (“don’t care”) bits and applies FEC

FEC Code Rate 1/2

Transmit 2 bits for each bit in original data stream:

Modulation	Code Rate	Data rate
BPSK	1/2	6 Mbit/s
QPSK	1/2	12 Mbit/s
16-QAM	1/2	24 Mbit/s

FEC Code Rate 3/4

Transmit 4 bits for each 3 bits in original data stream:

Modulation	Code Rate	Data rate
BPSK	3/4	9 Mbit/s
QPSK	3/4	18 Mbit/s
16-QAM	3/4	36 Mbit/s
64-QAM	3/4	54 Mbit/s

Additional 'stolen' FEC bits

802.11n adds code rate 5/6:

Modulation	Code Rate	Data rate
64-QAM	5/6	64 Mbit/s

Maximum data rate goes up from 58.5Mbit/s to 64Mbit/s.

Short Guard Interval (SGI)

The guard interval is a short period of “silence” between OFDM symbols. It avoids interference due to propagation delays, echos, and reflections.

- guard interval in 11a/b/g: 800ns
- 11n optionally supports guard interval of 400ns

With SGI, the maximum data rate goes up from 65 Mbit/s to 72.2Mbit/s. This is the maximum data rate for single-antenna devices on a 20MHz channel.

MIMO

Multiple Input Multiple Output (MIMO) radio systems use multiple antennas when sending and/or receiving data.

- The term “MIMO” does not describe how antennas are used
- 11a/b/g are SISO radio systems (Single Input Single Output). These never use more than one antenna simultaneously.[2]
- Assymetric configurations are possible: SIMO, MISO, and MIMO where one device has more antennas than the other.

[2] 11a/g antenna diversity selects just one of two antennas for receiving

MIMO

802.11n uses MIMO for:

- Spatial Multiplexing (higher data rates)
- Space-Time-Block-Coding (STBC) (better reception)
- Beamforming (larger range)

Spatial Multiplexing

Spatial Multiplexing divides data stream across multiple antennas.

- Transmitter has N antennas with sufficient physical distance between each other.
- Each antenna sends a subset of the data stream. All antennas send concurrently and on the same channel.
- Concurrent radio signals appear with unique signal propagation delays and echos at N antennas at the receiver.
- Powered by clever maths[3] the receiving PHY recognizes separate streams and derives the original data stream.

[3] indistinguishable from magic

Spatial Multiplexing

802.11n supports up to 4 pairs of transmit/receive antennas. Each pair adds throughput equivalent to SISO. The antenna configuration of a device is designated as nTmR (n transmit antennas, m receive antennas).

Antennas	Maximum data rate[4]
1T1R	72.2Mbit/s
2T2R	144.4Mbit/s
3T3R	216.7Mbit/s
4T4R	288.9Mbit/s

[4] for a 20MHz channel with SGI

Space-Time-Block-Coding (STBC)

STBC can be used in asymmetric configurations where transmitter has more antennas than receiver. It employs similar tricks as Spatial Multiplexing but provides for data stream redundancy and hence better reception.

- Transmitter sends redundant copies of data stream on N antennas.
- Redundant radio signals appear with unique signal propagation delays and echos at the receiver.
- Powered by clever maths the receiving PHY recognizes multiple copies of the stream and derives the original data stream.

Beamforming

Beamforming is used to steer radio signals towards the receiver, rather than radiating into all directions.

- Transmitter applies a phase-shift to the signals sent on each of its antennas. This affects the combined radiated signal lobe.
- By carefully adjusting its phase-shifting the transmitter can steer the lobe towards the receiver.
- Transmitter and receiver exchange non-data “sounding” frames to optimize beamforming parameters.
- The receiver experiences better signal-to-noise ratio.

40MHz channels

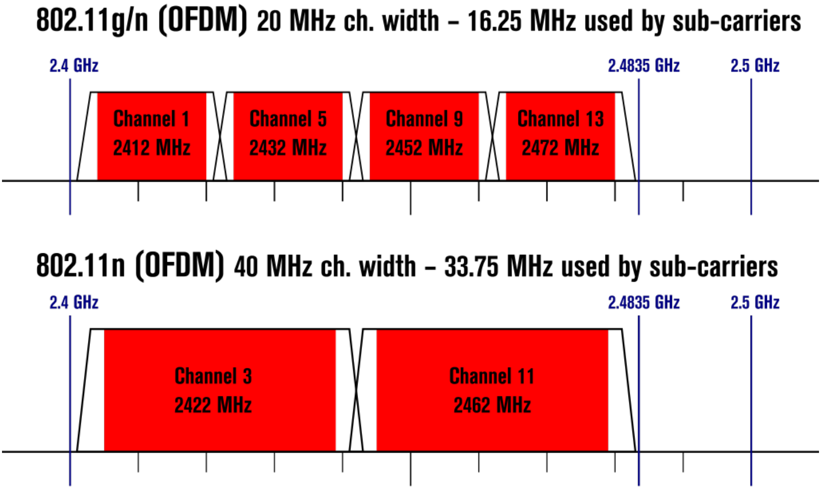
802.11n can join two adjacent 20MHz channels into a 40Mhz one.

- provides 114 subcarriers for OFDM, 108 used for data
- AP sends beacons on each 20MHz channel ("dual-beacon")
- AP manages dynamic co-existence of 20MHz-only and 40MHz-capable devices on overlapping channels

Antennas	Maximum data rate[5]
1T1R	150Mbit/s
2T2R	300Mbit/s
3T3R	450Mbit/s
4T4R	600Mbit/s

[5] for a 40MHz channel with SGI

40MHz channels in the 2.4Ghz band



Source: Wikipedia

40MHz channels in the 5Ghz band

The 5GHz band has a lot more space.

- 4 40MHz channels fit between channels 36 and 64
- 5 40MHz channels fit between channels 100 and 140
- outdoor channels require Dynamic Frequency Selection support[6] to prevent interference with radar

[6] there is no blob-free DFS implementation AFAIK

Modulation Coding Scheme (MCS)

MCS index describes a combination of:

- modulation scheme (BPSK, QPSK, QAM, ...)
- FEC code rate
- number of spatial streams (ie. antennas)

To know the data rate for a given MCS index the channel width and use of a short guard interval must also be known.
802.11n standard defines 77 MCS indices. Most vendors only implement up to MCS 31.

MCS 0-15

Modulation and coding scheme				Transmit data rate in Mbit/s			
				20 MHz channel		40 MHz channel	
MCS	Modulation	Code Rate	Antennas	GI	SGI	GI	SGI
0	BPSK	1/2	1	6.5	7.2	13.5	15.0
1	QPSK	1/2	1	13.0	14.4	27.0	30.0
2	QPSK	3/4	1	19.5	21.7	40.5	45.0
3	16-QAM	1/2	1	26.0	28.9	54.0	60.0
4	16-QAM	3/4	1	39.0	43.3	81.0	90.0
5	64-QAM	2/3	1	52.0	57.8	108.0	120.0
6	64-QAM	3/4	1	58.5	65.0	121.5	135.0
7	64-QAM	5/6	1	65.0[7]	72.2	135.0	150.0
8	BPSK	1/2	2	13.0	14.4	27.0	30.0
9	QPSK	1/2	2	26.0	28.9	54.0	60.0
10	QPSK	3/4	2	39.0	43.3	81.0	90.0
11	16-QAM	1/2	2	52.0	57.8	108.0	120.0
12	16-QAM	3/4	2	78.0	86.7	162.0	180.0
13	64-QAM	2/3	2	104.0	115.6	216.0	240.0
14	64-QAM	3/4	2	117.0	130.0	243.0	270.0
15	64-QAM	5/6	2	130.0[8]	144.4	270.0	300.0

[7] Any 802.11n device must support MCS 0-7 20MHz no SGI

[8] 802.11n access points must support MCS 0-15 20MHz no SGI

non-HT frame format

This is the frame format defined in 802.11a/g.

L-SFT	L-LTF	L-SIG	Data

- Legacy[9] Short Training Field
- Legacy Long Training Field
- Legacy Signal Field (specifies length of data part)

[9] "Legacy" nomenclature introduced in 802.11n

HT-mixed frame format

Backwards compatible with 802.11a/g since training and signal fields are identical.

L-SFT	L-LTF	L-SIG	HT-SIG	HT-STF	HT-LTF	...	HT-LTF	Data

- HT signal field contains information about MCS index, channel width, length of DATA part, use of SGI and MIMO.
- Short and long HT training fields are used with MIMO.

HT-greenfield frame format

Can only be used if all devices support 802.11n.
Optional, not required by 802.11n standard.

HT-GF-SFT	HT-LTF1	HT-SIG	HT-LTF	...	HT-LTF	Data

TX aggregation

Tx aggregation reduces overhead of medium contention. 802.11n defines two mechanisms for aggregating unicast data frames:

- Aggregate MAC Service Data Unit (A-MSDU) aggregate MSDUs (ie. frames with Ethernet headers)
- Aggregate MAC Protocol Data Unit (A-MPDU) aggregate MPDUs (ie. frames with 802.11 MAC headers)

A-MSDU

A-MSDU frame structure:

A-MSDU Subframe	A-MSDU Subframe	...	A-MSDU Subframe

up to 3839 or 7934 bytes total

Each subframe contains:

	Target Address	Source Address	MSDU Length	MSDU	Padding
Bytes	6	6	2	0-2304	0-3

The entire A-MSDU is acknowledged. All subframes are lost if a transmission error occurs in any part of the A-MSDU.

A-MPDU

A-MPDU frame structure:

A-MPDU Subframe	A-MPDU Subframe	...	A-MPDU Subframe

up to 65535 bytes total

Each subframe contains:

	Reserved	MPDU Length	CRC	Delimiter	MPDU[10]	Padding
Bytes	0.5	1.5	1	1	0-4095	0-3

Subframes are acknowledged with a 128 byte Block-Ack-Bitmap. Requires a Block-Ack agreement between sender and receiver. Receiver keeps track of failed A-MPDU subframes and delivers frames in the correct order by sequence number.

[10] the MPDU may be an A-MSDU

A-MPDU frame injection attack

The standard suggests an algorithm to find the next subframe after a CRC error. This algorithm is broken but is now deployed in various device firmware. Attack injects malicious A-MPDU subframes into WLAN and bypasses firewalls:

Reserved	MPDU Length	CRC	Delimiter	MDPU ... **hidden subframe** ...

- Hide some A-MPDU subframes in a large file.
- Trick user into downloading this file.
- Wait for a CRC error in data preceeding a hidden subframe.
- Firmware seeks forward and finds hidden subframe.
- Profit!

Workaround: Transmit A-MPDUs only if WPA encryption is active.

Source: `https://github.com/rpp0/aggr-inject`

Information Elements (IEs)

Information Elements are type-length-value fields in beacons, probe requests, probe responses, association requests, etc.
IEs can be displayed with tcpdump:

```
# tcpdump -n -i iwn0 -y IEEE802_11_RADIO -v

beacon, caps=2041<ESS,SHORT_PREAMBLE,SHORT_SLOTTIME>,
ssid (RadissonBlu Guest), rates 1M 2M 5M 11M, ds (chan 10),
tim 0x00010000, country ’SI ’, channels 1-13 limit 20dB,
erp 0x00, xrates 6M 9M 12M 18M 24M 36M 48M 54M,
vendor 0x0050f2020101800003a4000027a4000042435e0062322f00,
```

New IEs added in 802.11n

- HT capabilities element lists supported 802.11n features
- HT operation element lists settings which may change at runtime (e.g. channel width)

Increasing tcpdump's snapsize reveals them:

```
# tcpdump -n -i iwn0 -y IEEE802_11_RADIO -v -s 1500
beacon, ...
htcaps=<20MHz,LDPC,SGI@20MHz,TXSTBC,RXSTBC 1 stream,A-MSDU 3839,
A-MPDU max 65535,A-MPDU spacing 8.00us,RxMCS 0xffff0000000000000000>,
...
htop=<20MHz chan 10,STA chanw 20MHz,basic MCS set 0x0000000000000000>,
...
```

Initial 802.11n support in OpenBSD 5.9

Focus was on mandatory 11n features only. Development began at *c2k15* hackathon, Jul 15-21 2015, at SAIT, Calgary.

- show HTCAP and HTOP IEs in tcpdump(8)
- add 802.11n media type info to net/if_media.h
- show 11n media type info in ifconfig(8)
- support HTCAP and HTOP IEs in the kernel
- transmit frames with MCS 0-7 (20Mhz channel, no SGI)
- accept block ack agreements
- receive A-MPDUs
- receive A-MSDUs
- supported drivers: iwm(4), iwn(4)

Development concluded at *n2k15* hackathon, Dec 1-6 2015, Hannover.

The if_media word

32 bit if_media word in net/if_media.h:

Instance	Shared options	RFU	Options	Media type	Media subtype
31-28	27-20	19-16	15-8	7-5	4-0

Problem: Need 77 MCS in Media subtype field – not enough bits.
Solution: Time for an ABI break! Expand if_media word to 64 bits:

Instance	Shared options	RFU	Options	Media type	Media subtype
56-63	40-55	32-39	16-31	8-15	7-0

This change was made during the *l2k15* hackathon, Sep 8-13 2015, in Varaždin, Croatia.

The devil is in the details

- Must send a Microsoft WMM information element in probe and assoc requests to some APs to get 11n. Not mentioned in 802.11n standard.
- Our simplistic rate adaptation only deals in MCS, so tell firmware to retry frames at 11b rates in 2GHz networks.
- Block ack reordering buffer must be protected from sequence number jumps caused by buggy hardware/firmware. Every OS has hacks for this which the standard does not document.
- Initial block ack implementation examined sequence number of QoS no-data frames, which have an arbitrary number and confused the block ack reordering buffer.
- HT protection settings must be updated when changed by AP.
- WPA decryption drops replayed frames, but A-MPDU subframes may legitimately arrive out of order.

Changes made between OpenBSD 5.9 and OpenBSD 6.0

- netstat -W prints counters related to 802.11n events
- Improved Block Ack receive logic (joint work with tb@ at *p2k16*, Apr 25-29 2016, Nantes, France)
- support for 8260 and 3160 devices added to iwm(4)
- forced RTS/CTS protection on for large frames
- enabled HT protection updates in iwn(4)

Changes made since OpenBSD 6.0 in -current

- iwm(4) self-heals after fatal firmware errors
- prefer 5GHz APs over 2GHz ones with same ESSID
- iwm(4) driver code refactored to reduce size and complexity (at *g2k16*, Aug 30-Sep 5, 2016, Cambridge, UK)
- enabled SGI for iwm(4) and iwn(4) (also at *g2k16*)
- pass AP's contention window parameters to iwm(4) firmware, fixes timing of ACKs
- fixed transmit rates used for ACKs in iwm(4)

Planned changes for OpenBSD 6.1

- implement MIMO-aware transmit data rate adaptation which can switch between SISO MCS, MIMO MCS, and legacy 11a/b/g rates
- enable transmit at MCS 8 and above

Other unscheduled but desirable changes

- Tx aggregation (A-MPDU, A-MSDU)
- 11n support for more drivers, such as athn(4), ral(4), rtwn(4), urtwn(4), run(4), otus(4)
- 40 MHz channels
- 11n hostap support
- STBC and Beamforming

The End

Questions?

Please support OpenBSD developers and hackathons:
`https://www.openbsd.org/donations.html`