Accuracy Enhancements of the 802.11 Model and EDCA QoS Extensions in ns-3

Timo Bingmann

Decentralized Systems and Network Services Research Group
Institute of Telematics, University of Karlsruhe

February 6, 2009

DSN

Roadmap

ns-3 Introduction

ns-3 is

- a discrete-event network simulator.
- intended to replace ns-2.
- not backwards compatible to ns-2.

ns-3 Introduction

ns-3 Goals

- Create tools aligned with needs of modern networking research.
- Work as open-source project with active community participation.
- Improve repeatability of results in research papers.

ns-3 and ns-2

ns-3 is not based on ns-2:
drop ns-2's historic burdens.

- ns-3 is fully C++ .
- Leverage up-to-date features of C++ .
- Create optional language bindings like Python for interpreter frontends.

Design Patterns

Utilize modern design patterns in C++ :

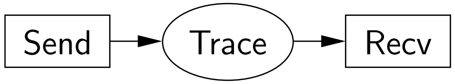
- Object and attribute system.
- Smart `Ptr<>` automatic memory management.
- Callbacks to decouple modules.
- COM-like object aggregation and interface querying.
- Decouple trace sources from sinks.

Requires advanced C++ knowledge.

Design Pattern: Tracing

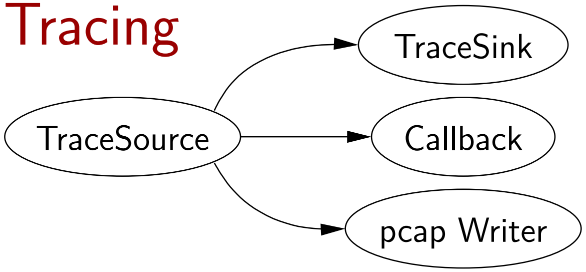
Tracing needs vary greatly in different simulations.

ns-2:



- Trace objects inserted as network elements.
- Fixed trace file format for further statistical processing.
- Not easily customizable to own experiment.
- Also available: queue monitors.

Design Pattern: Tracing



ns-3:

- Models export TraceSources.
  Examples: Node packet reception, 802.11 PHY state changes, TCP congestion window values.
- TraceSources can be connected to own callback functions
- or to predefined trace files generators for output in pcap/tcpdump format or ascii text.

	Existing core ns-2 models	Existing ns-3
Applications	ping, vat, telnet, FTP, HTTP, probabilistic and trace-driven traffic generators, webcache	OnOffApplication, asynchronous socket API, packet sockets
Transport layer	TCP (many variants), UDP, SCTP, XCP, TFRC, RAP Multicast: PGM, SRM, RLM	UDP, TCP
Network layer	Unicast: IP, MobileIP, generic distance vector and link state, IPinIP, source routing MANET: AODV, DSR, DSDV, TORA, IMEP	Unicast: IPv4, global static routing Multicast: static routing MANET: OLSR
Link layer	ARP, HDLC, GAF, MPLS, LDP, Diffserv MACs: CSMA, 802.11b, 802.15.4 (WPAN), satellite Aloha.	PointToPoint, CSMA, 802.11 MAC low, high and rate control algorithms
Physical layer	TwoWayGround, Shadowing, OmniAntennas, EnergyModel, Satellite Repeater	802.11a, Friis propagation loss, log distance loss, basic wired (loss, delay)
Core Support	RNGs, tracing monitors, mathematical support, test suite, animation (nam)	RNGs, unit tests, logging, callbacks, mobility visualizer

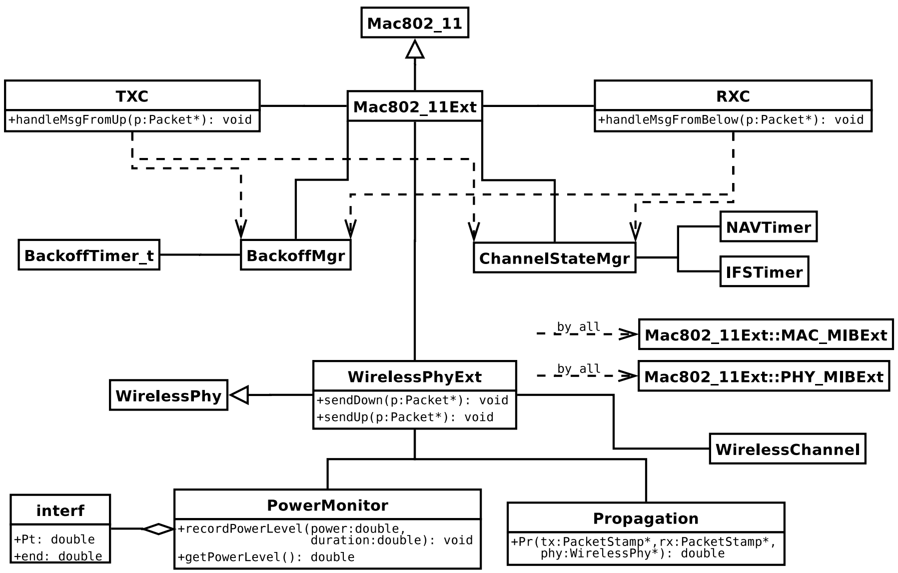
Based on tables from [1] and [2].

SLOC of ns-2.33 and ns-3.3

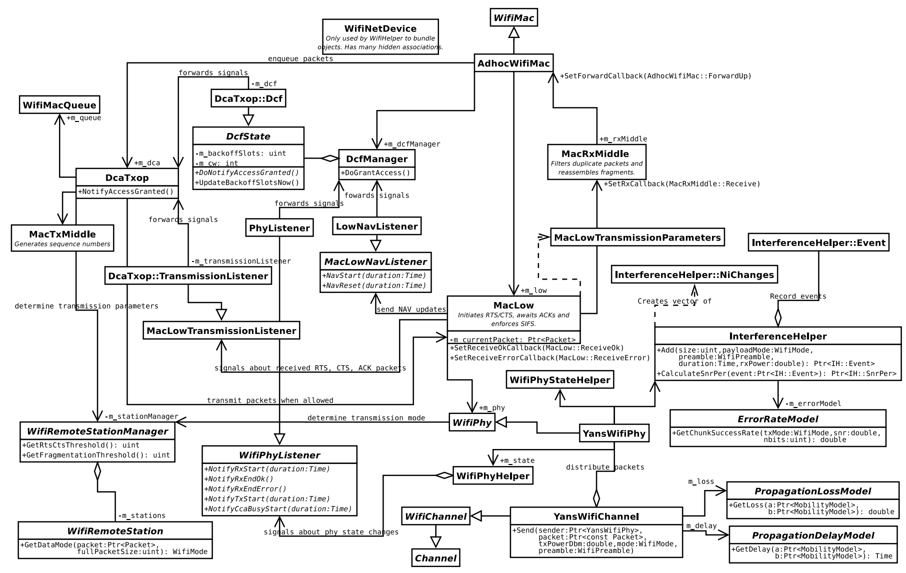
ns-2.33			ns-3.3		
C/C++	162,208	58%	C/C++	77,270	96%
Tcl	103,419	37%	Python[1]	2,906	4%
Other	13,341	5%	Other	58	0%
Total	278,968		Total	80,234	
802.11	6,067	2%	802.11	13,573	17%

[1] excludes automatically generated code
Statistics generated using David A. Wheeler's 'SLOCCount'.

UML of ns-2's Wifi Classes



UML of ns-3's Wifi Classes

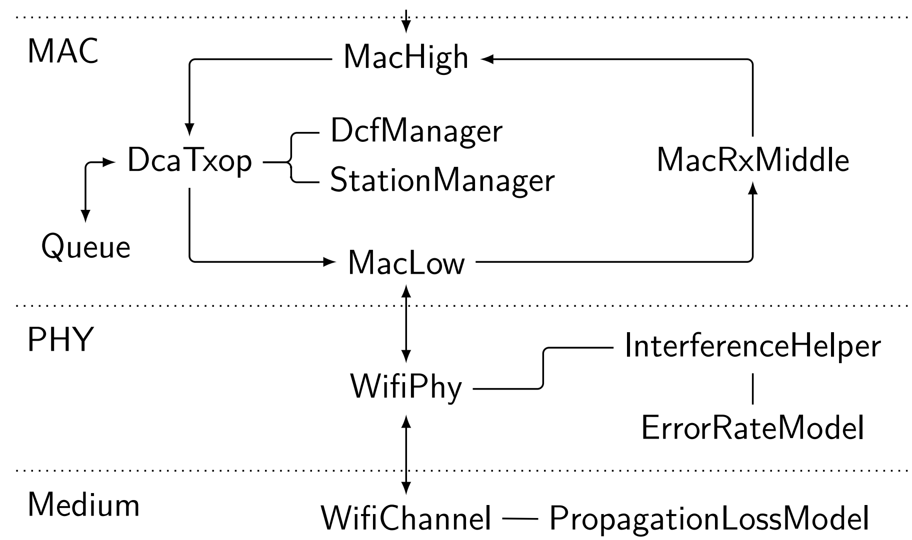


Thesis Goals

Goals

- ns-3 wireless simulations give equal or accountably different results like equivalent ns-2 simulations.
- Extend ns-3 with EDCA for 802.11e QoS.

Modelling 802.11 in ns-3



State of 802.11 in ns-3

PHY layer:

— Currently only 802.11a rates supported.
— No simulation of capture effect.
— No Nakagami propagation loss model.
+ BER/PER reception criterion.

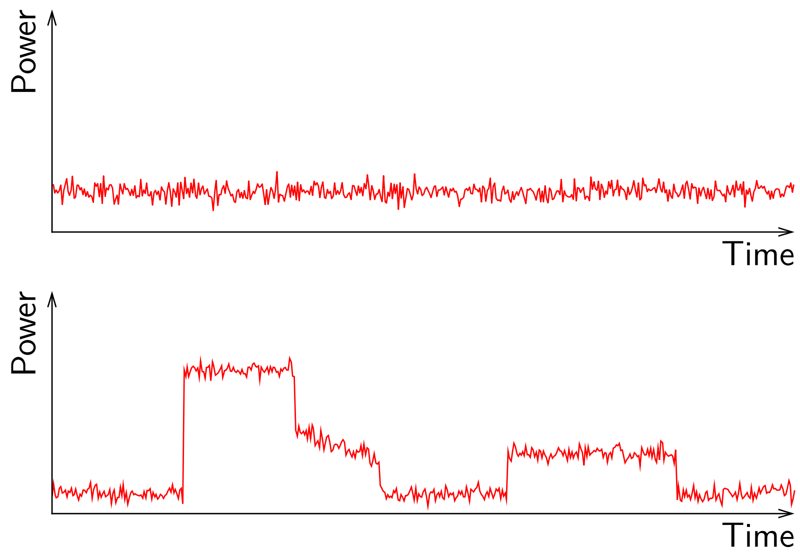
PHY Layer

Goal: compatibility with ns-2 WirelessPhyExt.

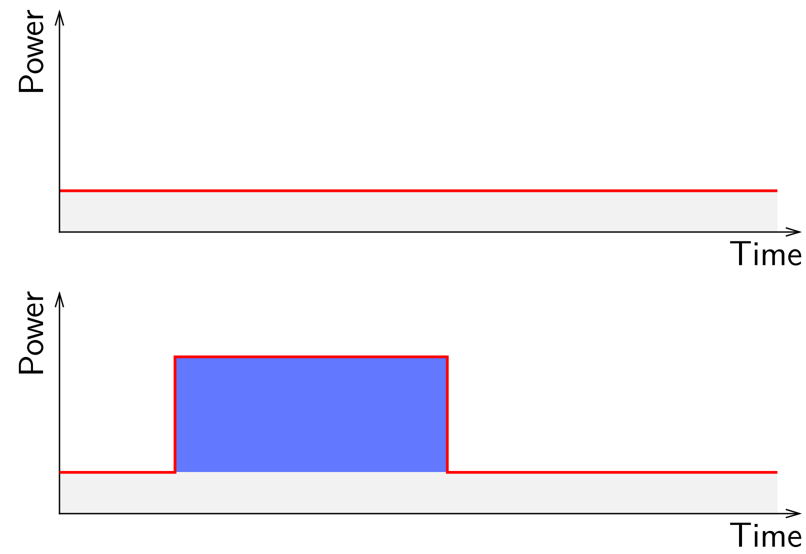
Required components

- PowerMonitor for cumulative noise
- SINR reception criterion
- Capture effect
- Nakagami propagation loss model

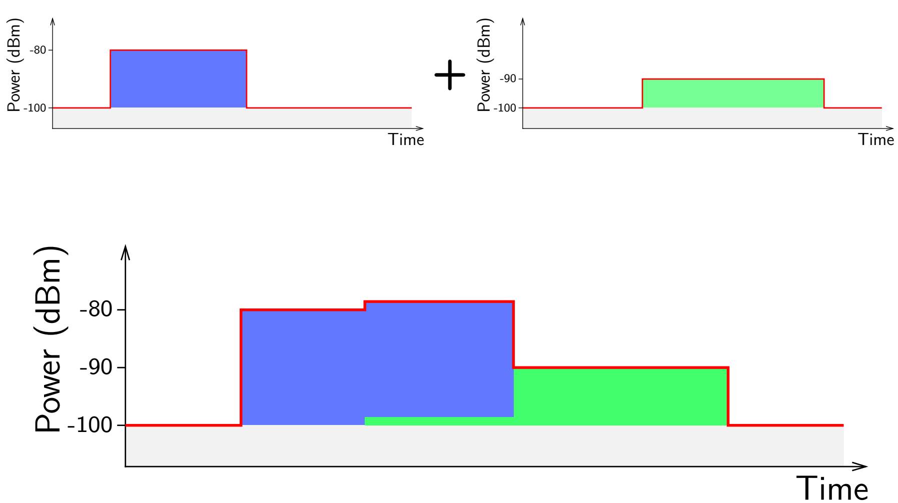
Signals, Noise and Interference



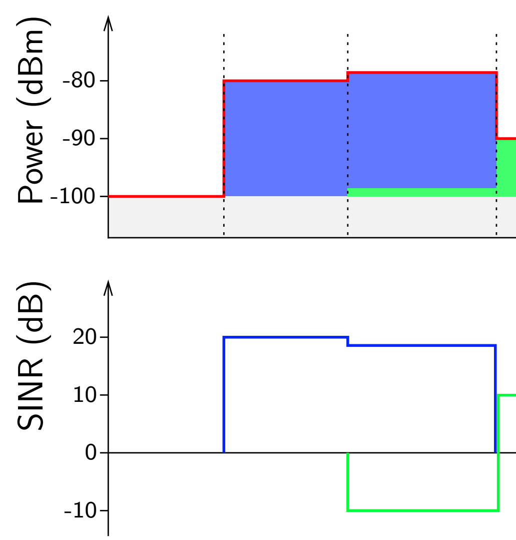
Signals, Noise and Interference



Signals, Noise and Interference

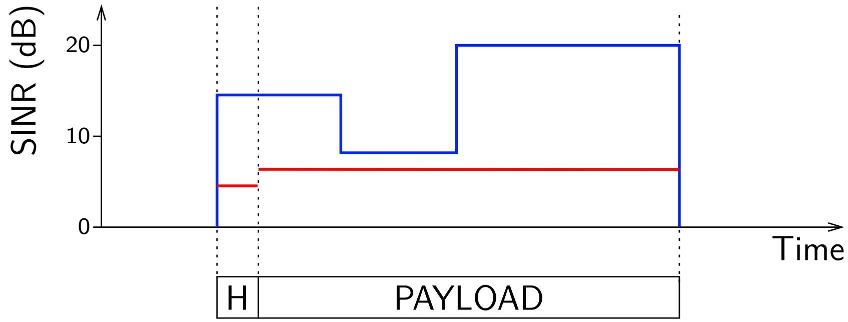


Signals, Noise and Interference



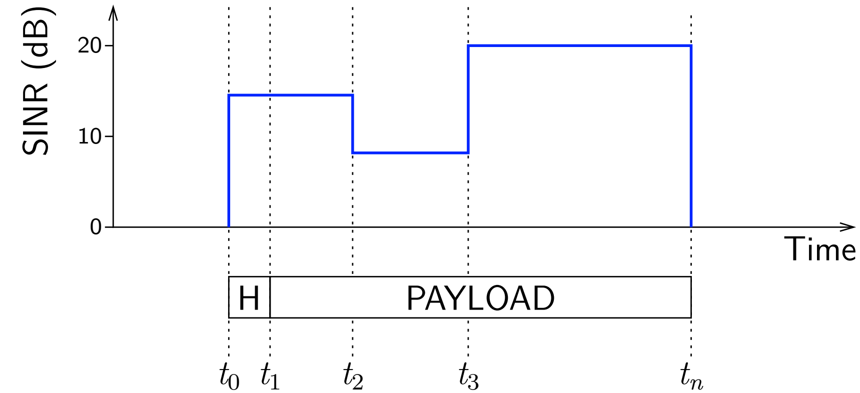
$$\text{SINR} = \frac{\text{Signal}}{\text{Noise} + \text{Interference}}$$

SINR Threshold



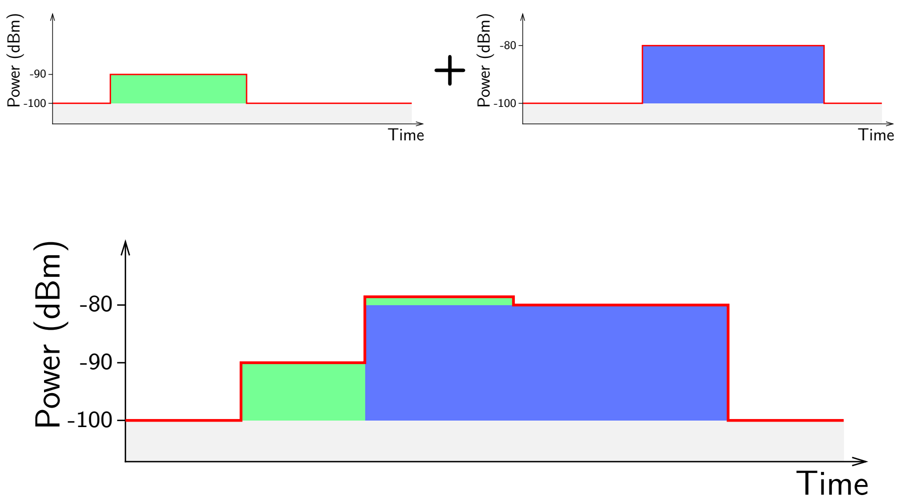
BPSK	5dB	QAM-16	15dB
QPSK	8dB	QAM-64	25dB

BER/PER Criterion

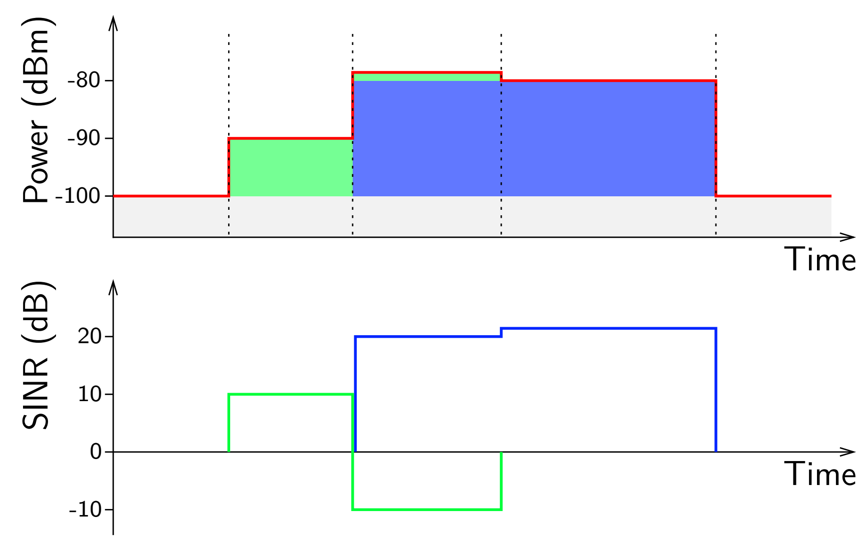


$$\mathrm{BER}_{\mathrm{BPSK}}\left(\frac{E_b}{N_0}\right) = Q\left(\sqrt{\frac{2E_b}{N_0}}\right)$$

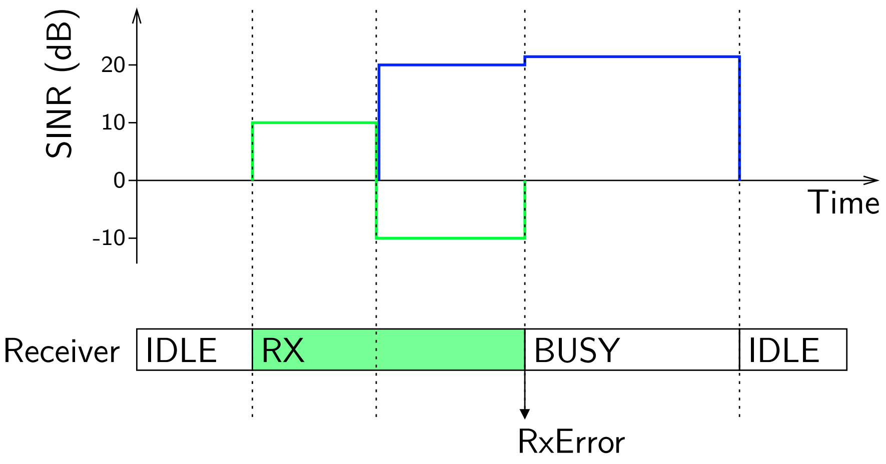
Capture Effect



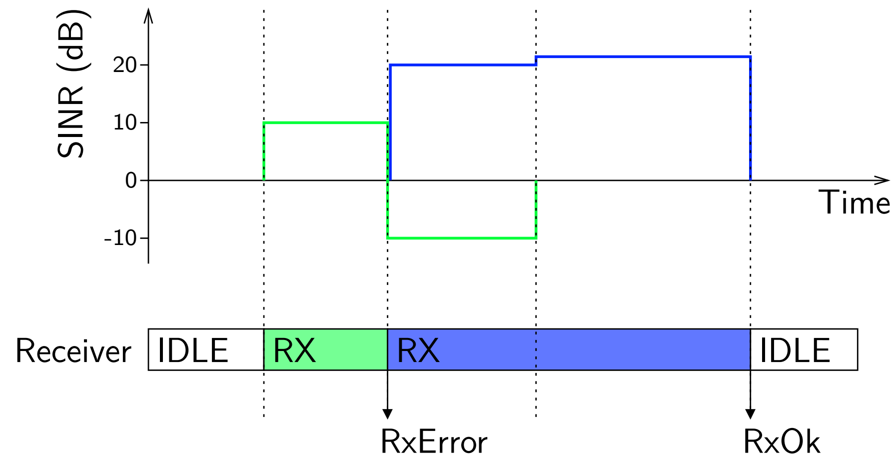
Capture Effect



Without Capture Effect



With Capture Effect

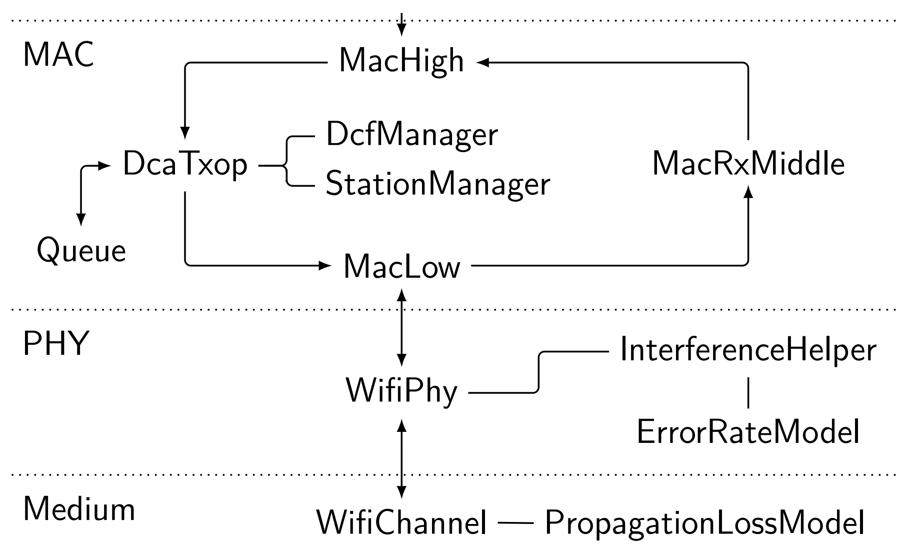


Thesis Goals

Goals

- ns-3 wireless simulations give equal or accountably different results like equivalent ns-2 simulations.
- Extend ns-3 with EDCA for 802.11e QoS.

Modelling 802.11 in ns-3



Short Recapitulation of DCF

Radio transmission using CSMA/CA:
Carrier sense multiple access with collision avoidance

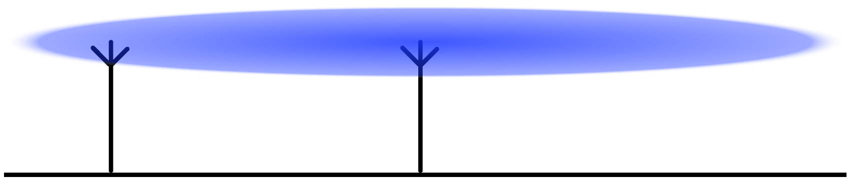
802.11 has two carrier sense mechansims:

- physical - CCA_BUSY
- virtual - NAV (network allocation vector)

Physical Carrier Sense

Stations always listen to the radio channel.

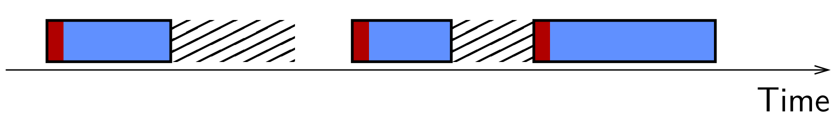
CCA_BUSY indication is raised
if radio energy level is above a CS threshold.



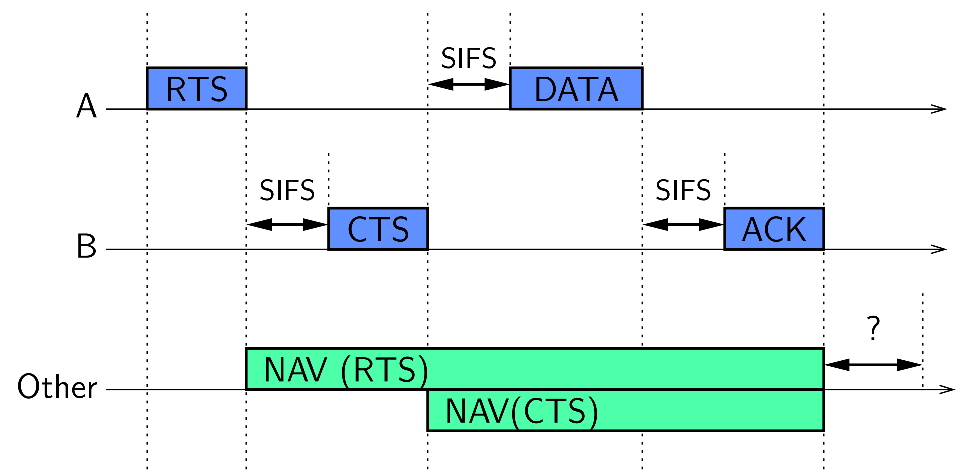
Virtual Carrier Sense

Stations hear and decode all packet headers on the radio channel.

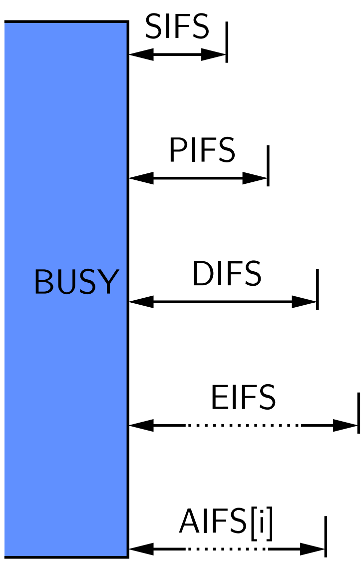
Header contains a duration field. Reserves channel for time after packet by updating NAV.



RTS/CTS using NAV



IFS - Interframe Spaces



Short IFS for direct answers to frame sequences.

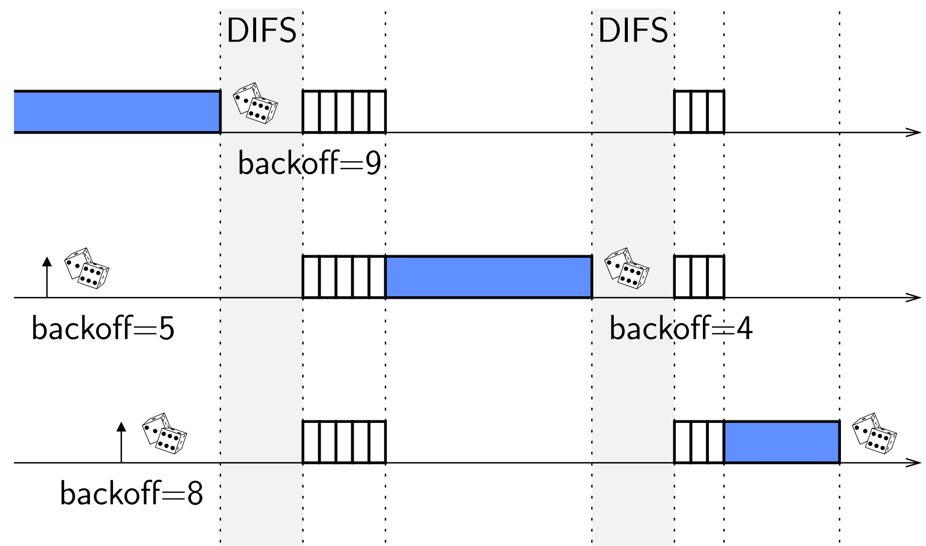
PCF IFS for Point Coordination Function

DCF IFS for Distributed Coordination Function

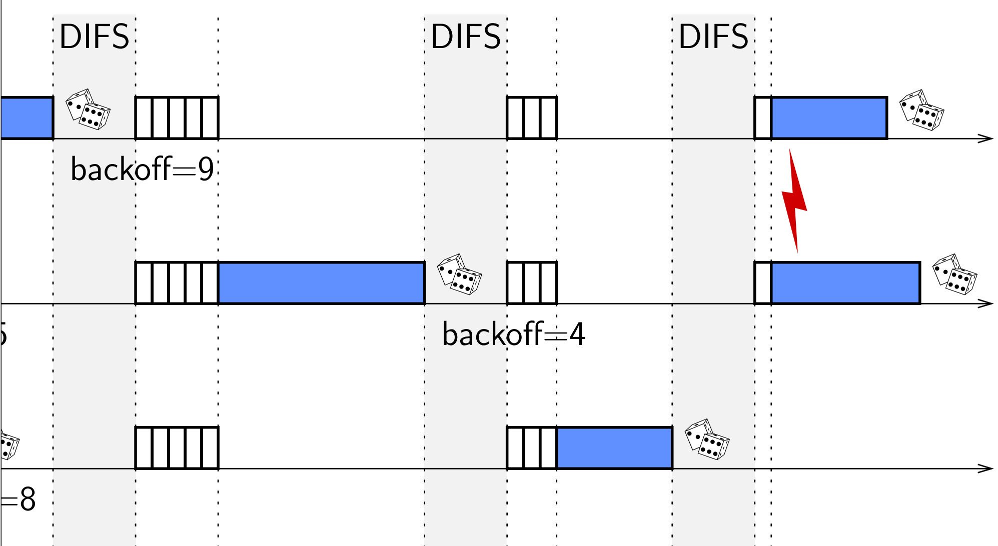
Extended IFS for error backoff.

Arbitration IFS for QoS.

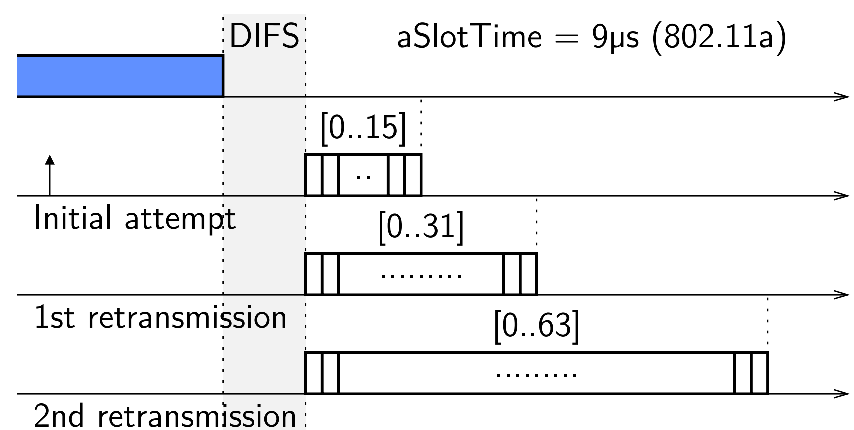
Backoff Procedure



Backoff Procedure



Contention Window



Backoff is uniform random integer from [0...CW].

Problems of DCF for QoS

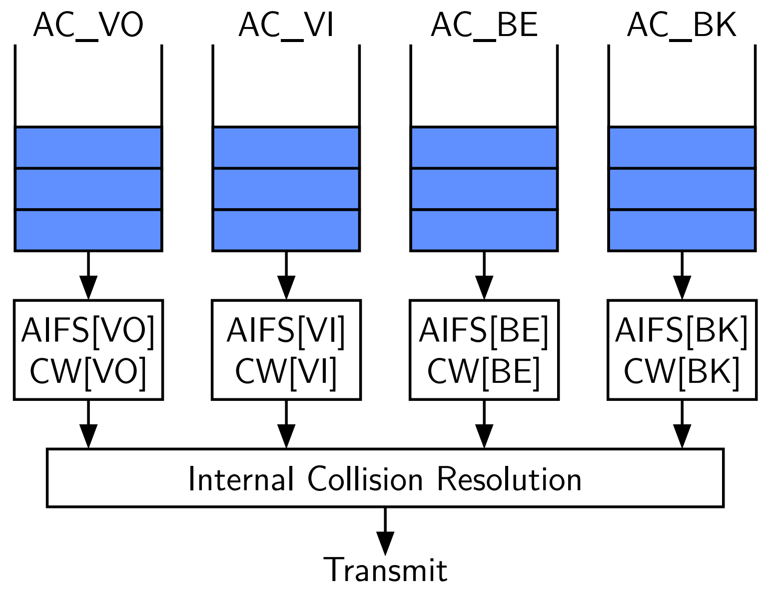
DCF is not good for time-critical traffic:

- Any STA may transmit arbitrarily large frames.
- All traffic stored in one queue.

PCF does not handles these issues:

- Contention-free period may be delayed.

EDCA Access Categories

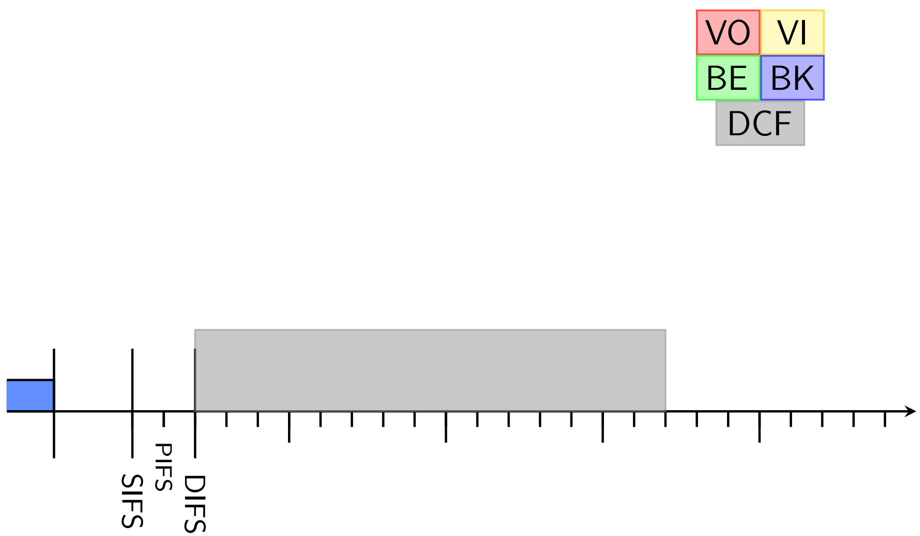


Default EDCA Parameters

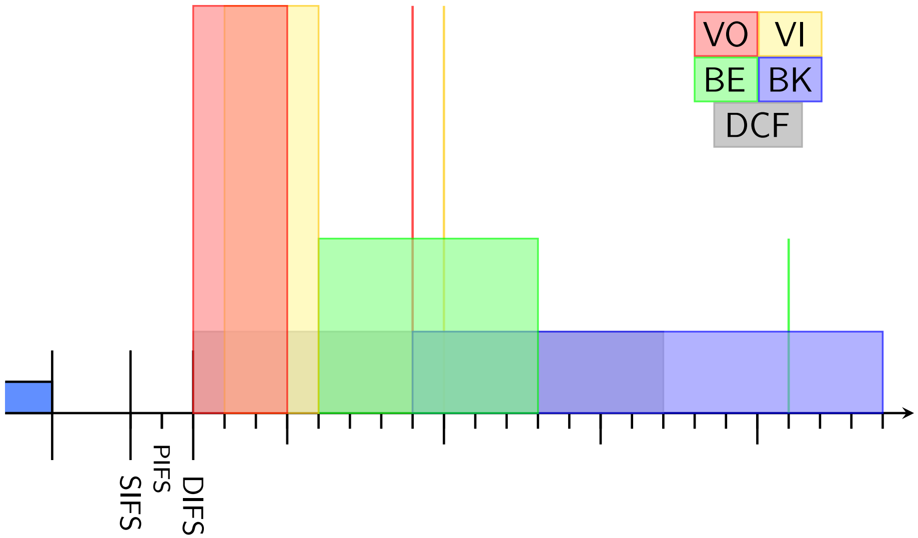
802.11p (Draft 4.02)

	VO	VI	BE	BK	DFS
CWmin	3	3	7	15	15
CWmax	7	7	15	1023	1023
AIFSN	2	3	6	9	2
AIFS	34µs	43µs	70µs	97µs	34µs

DCF Backoff Probability



Default EDCA Parameters of 802.11p



Work Status

Already finished:

- Ported NakagamiPropagationLossModel including dependencies.
- Implemented Ns2ExtWifiPhy for SINR reception and capture effect.

Outlook

Further Plans:

- Backport capture to BER/PER model.
- Implement and verify 802.11e EDCA QoS.
- Compilation and speed improvements with icc.
- Theoretical discussion of parallel or distributed 802.11 simulation.

Bibliography

[1] Thomas R. Henderson, Sumit Roy, Sally Floyd, and George F. Riley.
ns-3 project goals.
In *WNS2 '06: Proceeding from the 2006 Workshop on ns-2: the IP network simulator*, page 13, New York, NY, USA, 2006. ACM.

[2] Thomas R. Henderson.
ns-3 overview, December 2008.
`http://www.nsnam.org/docs/ns-3-overview.pdf`.