Cumulative Noise and 5.9 GHz DSRC Extensions for ns-2.28

Felix Schmidt-Eisenlohr, Marc Torrent-Moreno,
Tessa Tielert, Jens Mittag, Hannes Hartenstein

Institute of Telematics, University of Karlsruhe, Germany
{torrent, fschmidt}@tm.uni-karlsruhe.de,
ttielert@gmx.de, mittag@prime23.de,
hartenstein@rz.uni-karlsruhe.de

Technical Report 2006-21
ISSN 1432-7864
Dept. of Computer Science
Universität Karlsruhe (TH)

1 Introduction

The network simulator ns-2 [1] is a widely used tool in the field of wired and wireless communications research. Although the ns-2 implementation is in constant evolution, it fails, for obvious reasons, to reflect all aspects related to the newest communication technologies.

One example of such technology is 5.9 GHz Direct Short Range Communications (DSRC) [2] for vehicular environments, which is currently attracting attention due to its promises to reduce the amount of road fatalities and improve vehicular traffic efficiency on public roads. The IEEE 802.11p group is currently developing a standard to enable future 5.9 GHz DSRC based inter-vehicle communications.

In this report, we describe the modifications realized in ns-2 in order to model more accurately future wireless communications in vehicular environments. Section 2 reports the extensions performed to the PHY and MAC modules in order to include cumulative noise capabilities. Note that vehicular environments are specially sensible to the way interferences are modeled since safety related information will be commonly transmitted in a broadcast fashion. Section 3 describes the adjustments required to the MAC and PHY modules to reflect the behavior described in the current draft of IEEE 802.11p [3].

2 Cumulative Noise

In this section we describe the way we have modified the interference and reception model of ns-2 in order to provide better accuracy by introducing cumulative noise. The basis of our implementation is ns-2 in version number 28.

In general, the interference model describes how the simulator handles simultaneous signals corresponding to different transmissions at the wireless interface of a node. The reception model describes the way a simulator determines whether a packet can be successfully received or not.

Note that in this report we use the term *reception* for any signal that reaches a specific node, independently of the nodes' capability to successfully decode data from this signal, and the term *successful reception* for the capability to successfully decode the bits of a data packet from such a signal.

2.1 Default ns-2.28 interference and reception model

The wireless interface implemented in ns-2.28 is based on three power related thresholds:

- *Carrier Sense Threshold (CSTh)*: A message that arrives at a wireless interface of a node with a power lower than $CSTh$ will not be sensed. A message arriving with power equal to or higher than $CSTh$ is sensed by the wireless interfaces and the medium is determined *busy*.

- *Receiving Threshold (RxTh)*: A message arriving with a power equal to or higher than $RxTh$ in the absence of interferences can be successfully received.

- *Capture Threshold (CpTh)*: A message with power equal to or higher than $RxTh$ can be successfully received in the presence of interferences if the following two conditions are fulfilled:
  i) the packet arrives at the interface while the medium is *idle*
  ii) the power of the packet is $CpTh$ above the power of the strongest interference occurring during the reception of the packet[1].
  A packet arriving while the medium is already *busy* is never received successfully.

Additionally, the default implementation of ns-2.28 can only handle one incoming packet at a time, referred to in this document as the *current packet*. This way, at the moment that a packet is being received and a new one arrives a decision must be taken with respect to which packet, together with its corresponding information (received power, length, etc.), will be kept by the system and which one will be discarded.

This design implies some inaccuracies with respect to how packet reception and medium status detection is handled. With regard to condition *ii)* in the $CpTh$ definition we observe the following:

- If condition *ii)* is satisfied, the reception process is continued and the current packet is kept (*capture capability*), whereas the newly arriving one is discarded. If the newly arriving packet has a longer remaining reception time than the current packet, the medium status is indicated incorrectly at the end of a successful message reception, *idle* instead of *busy*.

- If the condition is not satisfied, both packets collide and none of them can be received successfully. The packet having the longer remaining reception time is kept, while the other one is discarded.

Note that the power of the current packet is compared to the power of each newly arriving packet individually, neglecting the effect that power levels of several of such packets can sum up to a higher interfering power level.

Summarized, the interference and reception model of the standard implementation leads to the following two statements:

- The interference model determines the state of the medium by comparing the signal power S of the packet that is currently being received with $CSTh$: it is determined *idle* if either no packet is currently being received at all, or

$$S < CSTh. \tag{1}$$

  It is determined *busy* if the following inequality holds for the signal power S of the received packet:

$$S \geq CSTh \tag{2}$$

- The reception model indicates a successful reception of a packet X with signal power S, if the medium was *idle* at the start of reception and, during the complete reception time of X, its reception power S is $CpTh$ above the maximum power of the interfering packets, I_{max}, arriving during its reception time, i.e., the following inequality is always satisfied:

$$S \geq I_{max} + CpTh \tag{3}$$

2.2 Extended interference and reception model

In our implementation, we have modified the way the reception of a signal is handled by a node with respect to the interferences by implementing cumulative noise capabilities. As depicted above, the original ns-2.28 code does not keep track of all ongoing messages at a node's interface, in particular it does not accumulate the power level of all ongoing interferences. In short words, we now consider the noise level together with all interfering signals as a cumulated noise level, that treats as the interference level when determining the successful reception of a message. We model this noise level as additive white Gaussian noise (AWGN) and accumulate all interfering signals by adding up their power values as it is already done by other network simulators like GloMoSim [4].

In the following we describe the derived state machine, see Figure 2. First of all we provide a set of necessary definitions for our model and describe the key points of our implementation.

[1] All absolute power values in this document are expressed in dBm, i.e., as a logarithmic ratio to a reception power of 1 mW. Ratios of power values in this document, like *CpTh*, are expressed in dB.

At each point in time *t* we define:

- *Node A*: A wireless node that receives one or more transmissions from other nodes.
- *Noise (N)*: The power level of the noise level (in dBm), i.e., a power level always being present and coming from natural sources, like thermal noise or radiation. N is set to a power level of -99 dBm due to private conversations with the company Siemens within the project NoW [5]. Transmissions with reception power lower than N can not be noticed and are directly discarded.
- *Signal (S)*: The power of the received signal at node A (in dBm), which corresponds to a packet transmission X from another node.
- *Interferences (I_S)*: The sum of the reception powers[2] of all noticeable transmissions except S, i.e., all transmissions with a reception power higher than N. Additionally, we call I the sum of the reception power of all noticeable transmissions, without regard to one specific signal S.
- *Cumulated Noise of S (CN_S)*: The aggregated power (in dBm) of all noise and interferences that are relevant for the signal S. CN_S is calculated as the sum of N and I_S. In accordance to the definition of I, we call CN the absolute cumulated power of all signals at node A and the noise level N.
- *Signal to Interference and Noise Ratio (SINR)*: The ratio of the signal S to its interferences CN_S. *SINR* is expressed in dB. Exemplarily, a *SINR* of 4 dB indicates that the value of S is 4 dB higher than the one of CN_S (in logarithmic representation); if values are expressed in Watts, S is a factor of $\sim$2.51 higher than CN_S.

Additionally we include in our model and implementation the *extended capture capability* that we have already described in [6]. It is implemented according to current wireless chipsets' capabilities [7]. The standard distribution of ns-2.28 only allows a message to be captured if it arrives at a moment when the medium is determined *idle*. With extended capture it is possible to successfully receive a message that arrives during a *busy* period of the medium, even when the interface has been receiving another message, as long as the *SINR* of the new message is equal to or higher than *CpTh*. According to private conversations with the company Siemens within the project NoW the newly arriving packet will not be received correctly if it arrives between 4 and 10 μs after the previous one due to resynchronization issues.

Using the given definitions and the extended capture capability, we now regard the thresholds introduced above in the following sense:

- *Carrier Sense Threshold (CSTh)*: The minimum absolute power value necessary for node A to detect the medium *busy*. $CSTh$ depends on the sensitivity of A's hardware.
- *Capture Threshold (CpTh)*: The minimum required *SINR* (in dB) that is necessary to successfully receive a packet X with signal power S, unless the resynchronization issues mentioned above prevent a successful reception. $CpTh$ depends on the modulation scheme that is utilized for the transmission of X.
- *Receiving Threshold (RxTh)*: The minimum reception power that is necessary for any successful reception of a packet X with signal power S in the absence of interferences I_s. According to the definitions above, $RxTh$ is $CpTh$ above N.

Concluding, the extended interference and reception model leads to the following two statements:

- The interference model determines the states of the medium by comparing CN and $CSTh$. It is determined *idle* if
$$CN < CSTh \tag{4}$$
and determined *busy* if
$$CN \geq CSTh. \tag{5}$$
- The reception model indicates the successful reception of a packet X with signal power S if, during the complete reception time of X, its *SINR* is equal to or higher than $CpTh$, i.e., the following inequality is always satisfied:
$$S \geq CN_S + CpTh \tag{6}$$

[2] Note that the sum of values given in dBm is defined as the arithmetic sum of the values' representation in mW.

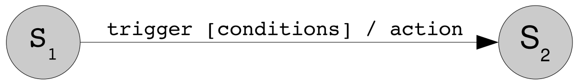

Figure 1: Generic structure of a transition.

Finite state machine

The state machine consists of a set of states, a set of possible state changes (transitions) and a set of interlinked actions that are expressed following the scheme shown in Figure 1.

Each state change consists of the following elements:

- the current state of the system S_1.
- the next state of the system S_2 in case the transition is taken.
- a transition between S_1 and S_2 that consists of:
  - a `trigger` that indicates the event that has to occur so that the transition is taken.
  - a set of `conditions` that have to be fulfilled so that the transition is taken.
  - an `action` that is taken after the transition.

In the following the components of the state machine shown in Figure 2 will be described.

States: The status of the wireless interface at a node A is described as a combination of a transmission and a reception state. The following three reception states are possible:

- *IDLE*: Node A currently does not sense any packet transmission on the medium, i.e., $CN < CSTh$.
- *BUSY*: Node A senses a signal on the medium, i.e., $CN \geq CSTh$, but a successful reception of a packet is not possible due to an earlier or currently detected violation of the conditions for successful packet reception defined in the extended interference and reception model.
- *RECV*: Node A senses a signal on the medium. Further, the power S of one packet X is higher than $RxTh$ and the reception conditions have been fulfilled up to now, i.e., *SINR*$(X) \geq CpTh$. Still, note that a later arriving packet could prevent the successful reception of X.

Two possible sending states exist:

- *noSEND*: Node A does not transmit a packet to the medium.
- *SEND*: Node A transmits a packet to the medium. In more detail, several sending states are differentiated in ns-2.28 depending on the type of packet that is transmitted: *SEND* for data packets, *RTS* for "Ready To Send" packets, *CTS* for "Clear To Send" packets, and *ACK* for "Acknowledgment" packets. To facilitate understanding and due to our focus on packet reception all these states are represented as one sending state *SEND*.

Trigger events of transitions: Possible trigger events are actions that can initiate a state change at node A. All triggers are associated with the start and the end of packet transmissions. A newly arriving packet is always expressed as X_n and has a reception power P_n. In case the node is in *RECV* state, the packet that is currently being received and has been successfully decoded so far is denoted as X_c and has a reception power P_c. The possible trigger events are:

- X_n: The first symbol of the preamble of X_n arrives at node A.
- X_c *end*: The last symbol of the currently received packet X_c has completely reached node A. Hence, the reception of X_c is finished.
- T_n: The first symbol of the preamble of a packet T_n is transmitted to the wireless channel by node A.
- T_c *end*: The last symbol of a packet T_c has completely been transmitted to the wireless medium by node A. Hence, the transmission is finished.
- $CN < CSTh$: The cumulated power level, CN, drops below $CSTh$ because the power of a packet, whose last symbol just arrived at node A but was not successfully received, is subtracted from CN. Hence, the channel is from now on determined *idle*.

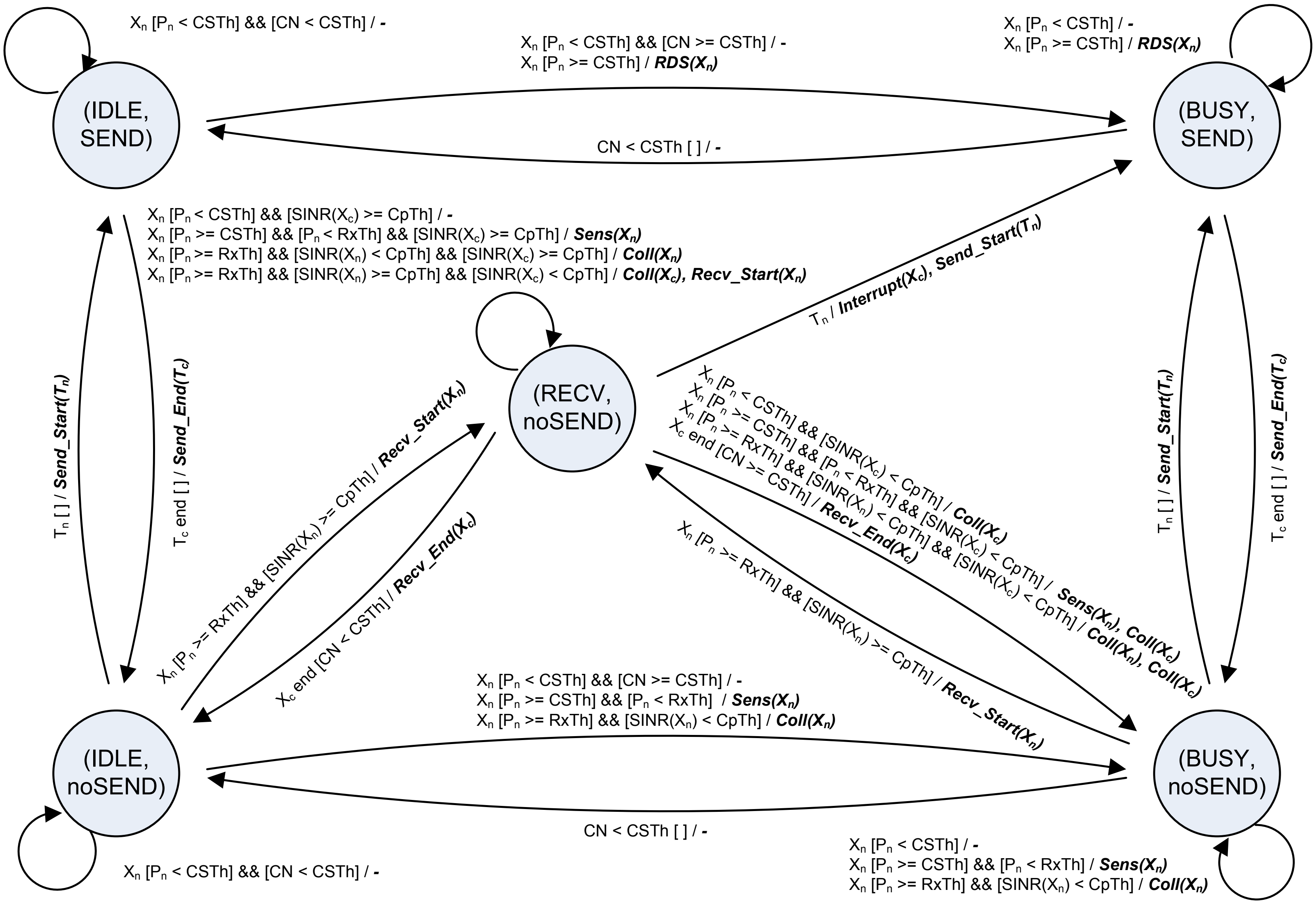

Figure 2: State machine of the wireless interface using the extended interference and and reception model for ns-2.28.

Conditions of transitions: Most trigger events are combined with one or several conditions that also have to be fulfilled so that the transition are taken. All abbreviations that are used have already been described in this report.

Actions of transitions: Transitions result in a specific action that is taken. The actions indicate how packets are classified in case the state of the interference and reception model has changed:

- $Recv\_Start(X_n)$: Node A starts to receive packet X_n.
- $Recv\_End(X_c)$: Node A has successfully finished to receive packet X_c.
- $Send\_Start(T_n)$: Node A starts to transmit packet T_n.
- $Send\_End(T_c)$: Node A has finished to transmit packet T_c.
- $Sens(X)$: Packet X is sensed by node A, but not received successfully due to too low reception power, i.e., its power is equal to or above $CSTh$, but below $RxTh$.
- $Coll(X)$: Packet X collides at node A and is not received successfully because its *SINR* has been smaller than $CpTh$ at any time during X's reception time.
- $RDS(X)$: Packet X is not received successfully at node A because A had been transmitting a packet to the medium when X arrived. Hence, X is "received during sending".
- $Interrupt(X)$: Packet x is not received successfully at node A because its reception has been interrupted by node A in order to transmit a packet to the medium. Note that the only case a packet reception is interrupted by a transmission is when an acknowledgment packet has been already scheduled and has to be sent immediately. Such packets are sent without regarding the medium state due to their importance for successful unicast packet transmissions (see the IEEE 802.11 Standard [8]).

2.3 Implementation

The extended interference and reception model described before is implemented by modifying and extending the current implementation of ns-2. We base our implementation on a version of ns-2.28 that includes the bug fixes published in [6]. Due to the structure of ns-2, most changes are done in the MAC implementation, i.e., in the file `ns-2.28/mac/mac-802_11.cc`.

Each packet X_n arriving at a node A with power P_n higher than N is taken into consideration. A new class *interference* is responsible for the correct handling of the current cumulative noise CN that is present at node A. CN is initialized with the power of the noise level, N. At each reception of a new packet X_n, the packet is given to this new class to update CN by adding P_n. Also, X_n is inserted into a list sorted by the transmission finish times and used to decrease CN by P_n again at the end of X_n's reception. The list is updated whenever the value of CN is requested. Then, a newly arriving packet is handled according to the transitions shown in the finite state machine, Figure 2.

2.4 Validation of the implementation

The finite state machine introduced above has been validated by setting up a table of all possible combinations of triggers and conditions for each state, eliminating non-feasible combinations and determining the finite state machine's transactions to the remaining ones. Non-feasible combinations have been identified by applying the following assumptions:

- $P_n < CSTh \Rightarrow P_n < RxTh$
- $P_n < RxTh \Rightarrow \mathit{SINR}(X_n) < CpTh$
- $P_n \geq CSTh \Rightarrow CN > CSTh$
- $BUSY \Rightarrow CN \geq CSTh$
- $CN \geq CSTh \Rightarrow \neg IDLE$
- $\mathit{SINR}(X_n) \geq CpTh \Rightarrow \mathit{SINR}(X_c) < CpTh$
- $\mathit{SINR}(X_c) \geq CpTh \Rightarrow \mathit{SINR}(X_n) < CpTh$

- $(P_n < RxTh)$ && $(P_n \geq CSTh) \Rightarrow SINR(X_n)$ does not influence the behavior of the state machine

Having systematically accounted for all possible combinations of states, triggers and conditions, we assume the finite state machine introduced in this report to be implemented correctly and completely.

As a second step in validation, debug messages have been added to the code in order to retrace the states and transitions of the underlying finite state machine. These messages are invoked in the code whenever a transition occurs and provide a snap-shot of the current state of the finite state machine and the conditions triggering the transition.

We use a validation script that automatically checks each transition and identifies operation errors. The script parses the generated debug output and identifies trigger, conditions and output for each transition. The script then checks whether this combination of values matches a correct transition. Furthermore, it assures that the end state of a transition matches the starting state of the next one and that the finite state machine handles and receives the correct packet if there is more than one.

3 5.9 GHz DSRC Extensions

The IEEE 802.11p draft [3] of the future standard introduces modifications mainly in the physical and management domain. Features like channel scanning or authentication and association procedures will not be used due to the safety nature of WAVE communications. On the other hand, elaborated channel management capabilites are envisioned due to its multi-channel approach. On the physical layer, the carriers use the 5.9 GHz range and channels are set to 10 MHz to reduce symbol interference. With respect to the basic channel access mechanisms of IEEE 802.11, i.e., the distributed coordination function (DCF) based on CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance), no changes are expected.

Since ns-2 does not implement management functions and our studies are focused on safety related information exchange, which is performed in one single channel, no modifications have been implemented in the management plane. Likewise, no extensions have been introduced on the channel access mechanisms apart from the bug fixes described in [6].

With respect to the physical layer, ns-2.28 models a Lucent WaveLAN 802.11 DSSS (Direct Sequence Spread Spectrum) radio interface. In order to model a WAVE OFDM system, which operates at 5.9 GHz with 10 MHz channels, several modifications were required according to IEEE 802.11a [9] and IEEE 802.11p [3].

Parameter	Value
Mac/802_11 CWMin_	15
Mac/802_11 CWMax_	1023
Mac/802_11 SlotTime_	16 μs
Mac/802_11 SIFS_	32 μs
Mac/802_11 PreambleLength_	32 μs
Mac/802_11 PLCPHeaderLength_	8 μs
Mac/802_11 PLCPDataRate_	3 Mbps
Mac/802_11 DataBitsPerSymbol_	24
Mac/802_11 RTSThreshold_	375 bytes
Mac/802_11 ShortRetryLimit_	7
Mac/802_11 LongRetryLimit_	4
Mac/802_11 basicRate_	3 Mbps
Mac/802_11 dataRate_	3 Mbps
Antenna/OmniAntenna Gt_	2.512
Antenna/OmniAntenna Gr_	2.512
Phy/WirelessPhy CPThresh_	4.0 dB
Phy/WirelessPhy CSThresh_	-96 dBm
Phy/WirelessPhy Noise_	-99 dBm
Phy/WirelessPhy RXThresh_	-95 dBm
Phy/WirelessPhy bandwidth_	3 Mbps
Phy/WirelessPhy freq_	5.9 GHz

Table 1: IEEE 802.11p settings for a data rate of 3 Mbps (BPSK modulation and 1/2 coding rate)

The preamble and the PLCP header are always transmitted using the most robust modulation scheme, Binary Phase Shift Keying (BPSK), and the lowest coding rate (1/2), which results in a basic rate of 3 Mbps, whereas the payload can be transmitted with a higher data rate. Consequently the duration of the preamble is doubled compared to IEEE 802.11a. According to the draft and in order to deal with the expected larger communication distances, SIFS and the slot time parameter as well as all parameters which depend on them are set to higher values. Note that 16 *service* bits of the PLCP header are transmitted with data rate, instead of the basic rate. Also note that padding bits are added in order to fill up the last symbol of a message.

Table 1 presents the main parameters configured in our version of the simulator for the most robust modulation, which provides a data rate of 3 Mbps. Table 1 shows the complete set of the parameters for ns-2.28 and their settings in accordance to the IEEE 802.11p standard. Note that the table includes two new parameters that are required for modeling OFDM and for setting the noise level, namely "Mac/802_11 DataBitsPerSymbol_" and "Phy/WirelessPhy Noise_".

References

[1] "Network Simulator ns-2," http://www.isi.edu/nsnam/ns/.

[2] "Dedicated Short Range Communications Project," http://www.leearmstrong.com/DSRC/DSRCHomeset.htm.

[3] "IEEE P802.11p/D0.21, Draft Amendment for Wireless Access in Vehicular Environments (WAVE)," Jan. 2006.

[4] "GloMoSim - Global Mobile Information Systems Simulation Library," http://pcl.cs.ucla.edu/projects/glomosim/.

[5] "The Network on Wheels Project," http://www.network-on-wheels.de.

[6] F. Schmidt-Eisenlohr, J. Letamendia-Murua, M. Torrent-Moreno, and H. Hartenstein, "Bug Fixes on the IEEE 802.11 DCF module of the Network Simulator ns-2.28," University of Karlsruhe, Tech. Rep. TR-2006-1, January 2006.

[7] A. Kochut, A. Vasan, A. Shankar, and A. Agrawala, "Sniffing Out the Correct Physical Layer Capture Model in 802.11b," in *Proceedings of the IEEE International Conference on Network Protocols (ICNP), Berlin, Germany*, Oct. 2004.

[8] "IEEE Std. 802.11-1999, Part11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications," IEEE Std. 802.11, 1999 edition.

[9] "IEEE Std 802.11a-1999, Amendment 1: High-Speed Physical Layer in the 5 GHz Band," 1999.