Towards High Performance Modeling of the 802.11 Wireless Protocol

Luiz Felipe Perrone
Institute for Security Technology Studies
Dartmouth College

With Jason Liu, David Nicol and Michael Liljenstam
Network Protocol Modeling

Question 1: Why construct simulation models of network protocols?

Answer: Validation, verification, and performance evaluation.

Question 2: How much detail does one need to include in a simulation model?

Answer: Only as much as necessary. No more, no less.
Detail: When Enough is Enough

If you know what you want to study, you can judiciously weed out the meaningless detail from your simulation model.

We want to study **routing** in *ad hoc* networks; we don’t need to be overly concerned with the fine details in its MAC layer, as long as we can approximate its behavior.
A Little Taste of the 802.11 MAC Protocol

DCF Basic Access Method (BAM):

Station A

Station B

DIFS
DATA
SIFS
ACK
A Little Taste of the 802.11 MAC Protocol (cont.)

DCF Basic Access Method (BAM):

Simple, yes, but think about how many simulation events this translates into…
The Behavior of 802.11

- Messages are queued for TX until we run out of buffer space and, after that, simply dropped.
- What counts are the transmissions that each receiver can hear.
- We can approximate the sensing and the interfering ranges of each receiver.
- Most importantly, the 802.11 RF channel seems to have some kind of memory...
Modeling Goals

- Replicate the lossy behavior (due to queueing and channel limitation).
- Quantify message service time (transmission plus acknowledgement) from the channel state.
- Avoid transitional state: consider the channel at low utilization or at saturation.
Model Development

- We took the detailed 802.11 model from GloMoSim and ported it to SWAN, our Simulator for Ad Hoc Networks (currently under development).

- We ran simulations with 802.11 and studied how it performs for increasing offered load. We also determined message service times as function of channel state and number of stations.
If a transmitter is within sensing range of the receiver, its messages are accepted. Otherwise, they are immediately thrown away.

Cutoff $\rightarrow$ model scalability $\rightarrow$ efficient parallelization.
\( \delta_k = \) Elapsed time between tx of messages k and (k-1)

If message length is exponentially dist. with mean \(1/\lambda\), then

\[
\Pr\{\text{msg sent } \delta_k \text{ ago is still in channel}\} = e^{-\lambda \delta_k}
\]

Channel busy-ness: model the number of “active” messages in the channel.

\[
B_k(\delta_k) = 1 + e^{-\lambda \delta_k} B_{k-1}(\delta_{k-1})
\]
The general shape of the throughput curve for CSMA + some mathematical manipulation is given by:

\[ B_k(\delta_k) = 1 + e^{-\lambda \delta_k} B_{k-1}(\delta_{k-1}) \]

The loss model is now simple: at the receiver, throw a \( Bernoulli(P_{k}^{\text{loss}}) \) to determine if a message is to be successfully received or discarded (lost).
A Study in Model Validation

- We implemented our simple mode in SWAN.

- We ran simulations (with the same input parameters) of both the detailed and simple models and compared the results.

- The constants in the simple model had to be fine tuned to match the behavior of the detailed model.

- Outside the transition stage (offered loads between 70 and 85% of maximum capacity), we observed a good match.
Good News

- Experiments on a single P3 750MHz (Linux) with 512Mb; simulation length of 10,000s.
- SWAN runs on the DaSSF simulation engine.
- Single timeline.

<table>
<thead>
<tr>
<th>Stations</th>
<th>Load</th>
<th>Speedup</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>10%</td>
<td>1.78</td>
</tr>
<tr>
<td>5</td>
<td>50%</td>
<td>2.00</td>
</tr>
<tr>
<td>5</td>
<td>90%</td>
<td>2.78</td>
</tr>
<tr>
<td>10</td>
<td>10%</td>
<td>2.0</td>
</tr>
<tr>
<td>10</td>
<td>50%</td>
<td>2.18</td>
</tr>
<tr>
<td>10</td>
<td>90%</td>
<td>3.26</td>
</tr>
<tr>
<td>20</td>
<td>10%</td>
<td>2.21</td>
</tr>
<tr>
<td>20</td>
<td>50%</td>
<td>2.36</td>
</tr>
<tr>
<td>20</td>
<td>90%</td>
<td>3.73</td>
</tr>
</tbody>
</table>
The Road Ahead

- The simple model enhances lookahead for parallel simulation by a factor of 1,500.
- Initial parallel experiment with a 5 station model (each one on its own timeline), the simple model exhibits speedup of 250 (relative to the detailed model in one timeline).
- The simple model indicates potential for substantial lookahead improvement.