ns-3 Introduction

ns-3 is

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

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 Basics

1.1 Introduction

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.

802.11 Enhancements in ns-3

Timo Bingmann - 5/44
University of Karlsruhe

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.

1.2 Showcase: Design Patterns

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.

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.
1.3 Current State

Existing core ns-2 models

- ping, vat, telnet, FTP, HTTP, probabilistic and trace-driven traffic generators, webcache

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

Network layer

- Unicast: IP, MobileIP, generic distance vector and link state, IPinIP, source routing
- MANET: AODV, DSR, DSDV, TORA, IMEP

Link layer

- ARP, HDLC, GAF, MPLS, LDP, DiffServ
- Multicast: static routing
- MANET: OLSR

Physical layer

- TwoWayGround, Shadowing, OmniAntennas, EnergyModel, Satellite Repeater

Core Support

- RNGs, tracing monitors, mathematical support, test suite, animation (nam)
- RNGs, unit tests, logging, callbacks, mobility visualizer

SLOC of ns-2.33 and ns-3.3

<table>
<thead>
<tr>
<th></th>
<th>ns-2.33</th>
<th></th>
<th>ns-3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/C++</td>
<td>162,208</td>
<td>58%</td>
<td>77,270</td>
</tr>
<tr>
<td>Tcl</td>
<td>103,419</td>
<td>37%</td>
<td>2,906</td>
</tr>
<tr>
<td>Other</td>
<td>13,341</td>
<td>5%</td>
<td>Other</td>
</tr>
<tr>
<td>Total</td>
<td>278,968</td>
<td></td>
<td>Total</td>
</tr>
</tbody>
</table>

802.11 Enhancements in ns-3

802.11 Enhancements in ns-3

<table>
<thead>
<tr>
<th></th>
<th>ns-2.33</th>
<th></th>
<th>ns-3.3</th>
</tr>
</thead>
<tbody>
<tr>
<td>802.11</td>
<td>6,067</td>
<td>2%</td>
<td>802.11</td>
</tr>
</tbody>
</table>

1 excludes automatically generated code

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

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.

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

\[ \text{SINR} = \frac{\text{Signal}}{\text{Noise} + \text{Interference}} \]
SINR Threshold

<table>
<thead>
<tr>
<th>Modulation</th>
<th>SINR (dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK</td>
<td>5 dB</td>
</tr>
<tr>
<td>QAM-16</td>
<td>15 dB</td>
</tr>
<tr>
<td>QPSK</td>
<td>8 dB</td>
</tr>
<tr>
<td>QAM-64</td>
<td>25 dB</td>
</tr>
</tbody>
</table>

BER/PER Criterion

\[
BER_{BPSK} \left( \frac{E_b}{N_0} \right) = Q \left( \sqrt{\frac{2E_b}{N_0}} \right)
\]

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 mechanisms:
- 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**

- Stations A and B exchange RTS and CTS frames.
- NAV (RTS) and NAV (CTS) periods define the channel reservation.
- ACK frames follow the DATA frames.
**IFS - Interframe Spaces**

- **SIFS**: Short IFS for direct answers to frame sequences.
- **PIFS**: PCF IFS for Point Coordination Function
- **DIFS**: DCF IFS for Distributed Coordination Function
- **EIFS**: Extended IFS for error backoff.
- **AIFS[i]**: Arbitration IFS for QoS.

**Backoff Procedure**

- **DIFS**: backoff=9:
- **DIFS**: backoff=5:
- **DIFS**: backoff=4:
- **DIFS**: backoff=8:

**Contention Window**

- **aSlotTime = 9μs (802.11a)**
- Initial attempt: [0..31]
- 1st retransmission: [0..63]
- 2nd retransmission: [0..127]

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 handle these issues:
- Contention-free period may be delayed.

Default EDCA Parameters

802.11p (Draft 4.02)

<table>
<thead>
<tr>
<th>Access Category</th>
<th>CWmin</th>
<th>CWmax</th>
<th>AIFS</th>
<th>AIFSN</th>
</tr>
</thead>
<tbody>
<tr>
<td>VO</td>
<td>3</td>
<td>7</td>
<td>34μs</td>
<td>2</td>
</tr>
<tr>
<td>VI</td>
<td>3</td>
<td>7</td>
<td>43μs</td>
<td>3</td>
</tr>
<tr>
<td>BE</td>
<td>7</td>
<td>15</td>
<td>70μs</td>
<td>6</td>
</tr>
<tr>
<td>BK</td>
<td>15</td>
<td>1023</td>
<td>97μs</td>
<td>9</td>
</tr>
<tr>
<td>DFS</td>
<td>15</td>
<td>1023</td>
<td>34μs</td>
<td>2</td>
</tr>
</tbody>
</table>
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
