Performance Analysis of ARQ Protocols using a Theorem Prover

Osman Hasan  Sofiene Tahar

Hardware Verification Group
Concordia University
Montreal, Canada

ISPASS 2008
Objectives

- Probabilistic Theorem Proving
  “A robust and precise probabilistic analysis technique”

- What is it?
- Why do we need it?
- How can we apply it for the performance analysis of ARQ Protocols?
Outline

- Introduction
- Theorem Proving based Performance Analysis
- Performance Analysis of ARQ Protocols
- Conclusions
Motivation

Performance Analysis

- Simulation
  - State-of-the-art
  - Inaccurate results

- Theorem Proving
  - Proposed Solution

- Environmental Conditions
- Probabilistic Choice
- Noise
- Aging Phenomena
- Unpredictable Inputs

Theorem Proving
Performance Analysis of ARQ Protocols using Theorem Proving

Random Components

Hardware
Software

System Model
Random Variables (Discrete/Continuous)

Computer Based Analysis Framework

Property Satisfied?

Probabilistic and Statistical Properties
## Probabilistic Analysis Approaches

<table>
<thead>
<tr>
<th></th>
<th>Simulation</th>
<th>Formal Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Model Checking</td>
</tr>
<tr>
<td><strong>Random Components</strong></td>
<td>Approximate random variable functions</td>
<td>Probabilistic State Machine</td>
</tr>
<tr>
<td><strong>Analysis</strong></td>
<td>Observing some test cases</td>
<td>Exhaustive Verification</td>
</tr>
<tr>
<td><strong>Accuracy</strong></td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Expressiveness</strong></td>
<td>✓</td>
<td>✗</td>
</tr>
<tr>
<td><strong>No CPU Time Issue</strong></td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td><strong>Automation</strong></td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Theorem Prover

- A notation (syntax)
- A small set of fundamental axioms (facts)
  - A Boolean variable can be True or False: ∀ a.(a = T) ∨ (a = F)
- A small set of inference (deduction) rules
  - Equality is transitive: ∀ a b c. (a = b) ∧ (b = c) ⇒ (a = c)

Soundness
- Every new theorem must be created from
  - Basic axioms and primitive inference rules
  - Already proved theorems or inference rules

Theory (collection of verified theorems in a file)
- Can be reloaded in theorem provers
- Facilitates the instant utilization of already verified theorems
### Theorem Proving - Example

- Check if \( y > x \) for the given system (\( x \) is a natural number)

\[
x \rightarrow (x + 1)^2 \rightarrow y
\]

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>Problem statement</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>y &gt; x</td>
<td>Problem statement</td>
</tr>
<tr>
<td>2</td>
<td>((x+1)^2 &gt; x)</td>
<td>Implementation</td>
</tr>
<tr>
<td>3</td>
<td>((x+1)(x+1) &gt; x)</td>
<td>Definition of Square</td>
</tr>
<tr>
<td>4</td>
<td>((x+1)x + (x+1) \cdot 1 &gt; x)</td>
<td>Distributivity</td>
</tr>
<tr>
<td>5</td>
<td>(x \cdot x + 1 \cdot x + 1 \cdot 1 &gt; x)</td>
<td>Distributivity</td>
</tr>
<tr>
<td>6</td>
<td>(x \cdot x + x + x + 1 &gt; x)</td>
<td>Multiplicative Identity</td>
</tr>
<tr>
<td>7</td>
<td>(x \cdot x + x + 1 + x &gt; x)</td>
<td>Additive Commutativity</td>
</tr>
<tr>
<td>8</td>
<td>(x \cdot x + x + 1 &gt; 0)</td>
<td>Addition Cancellation</td>
</tr>
<tr>
<td>9</td>
<td>True</td>
<td>Natural numbers &gt; 0</td>
</tr>
</tbody>
</table>
Outline

- Introduction
- Theorem Proving based Performance Analysis
- Performance Analysis of ARQ Protocols
- Conclusions
HOL Theorem Prover

- Higher-order logic theorem prover
  - University of Cambridge, UK
  - 5 axioms
  - 8 primitive inference rules

Numerous proof assistants are available

Inbuilt mathematical theories of Boolean, list, set, integers, real analysis, measure, and probability theory
Theorem Proving Based Performance Analysis

System Description

System Properties

Random Components

Discrete Random Variables

Continuous Random Variables

Probabilistic Properties

Statistical Properties

Probabilistic Analysis Theorems

Theorem Prover

Formal Proofs of Properties

System Properties (Discrete Random Variables)

System Properties (Continuous Random Variables)
Formal Verification of Random Variables

- Measure Theory
- Probability space of Infinite Boolean sequence ($\mathbb{B}^\infty$)

$$\mathbb{B}^\infty : \text{positive integers} \rightarrow \text{Boolean}$$

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>T/F</td>
<td>T/F</td>
<td>T/F</td>
<td>T/F</td>
<td>T/F</td>
<td>T/F</td>
<td>T/F</td>
<td>T/F</td>
</tr>
</tbody>
</table>

- A random variable that
  - Accepts: $\alpha$
  - Returns: $\beta$

can be modeled in HOL as a function

$$f : \alpha \rightarrow \mathbb{B}^\infty \rightarrow (\beta \times \mathbb{B}^\infty)$$
Random Variables in HOL

Example

- **Coin Flip (Head, Tail)**

\[ B^{\infty} \rightarrow (\text{flip\_outcome} \times B^{\infty}) \]

- **Algorithm**

  \[
  \text{flip } s = \\
  \begin{cases} \text{Head} & \text{if (top element of } s\text{) then} \\
  \text{Tail} & \text{else Tail,} \\
  \text{remaining portion of } s \end{cases}
  \]

- **Probabilistic Properties**

  \[
  P\{s \mid \text{flip } s = \text{Head}\} = \frac{1}{2}
  \]
# Discrete Random Variables in HOL

## Theorems: Discrete Random Variables

<table>
<thead>
<tr>
<th>Random variable</th>
<th>HOL Function</th>
<th>PMF (Pr (X = n))</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform(m)</td>
<td>unif_rv</td>
<td>$\frac{1}{m}$</td>
</tr>
<tr>
<td>Bernoulli(p)</td>
<td>bern_rv</td>
<td>$p$</td>
</tr>
<tr>
<td>Geometric(p)</td>
<td>geom_rv</td>
<td>$p(1 - p)^n$</td>
</tr>
</tbody>
</table>
## Continuous Random Variables in HOL

<table>
<thead>
<tr>
<th>Random Variable</th>
<th>HOL Functions</th>
<th>CDF (Pr (X ≤ x))</th>
</tr>
</thead>
</table>
| Exponential(l)   | exp_rv              | \[
\begin{cases}
0, & x \leq 0 \\
1 - \exp^{-lx}, & 0 < x
\end{cases}
\] |
| Uniform(a,b)     | uniform_rv          | \[
\begin{cases}
0, & x \leq a \\
\frac{x - a}{b - a}, & a < x \leq b \\
1, & b < x
\end{cases}
\] |
| Rayleigh(l)      | rayleigh_rv         | \[
\begin{cases}
0, & x \leq 0 \\
\frac{-x^2}{2l^2}, & 0 < x
\end{cases}
\] |
Verification of Statistical Properties

Definition: Expectation for Discrete Random Variables

\[ Ex[X] = \sum_{i=1}^{\infty} i \Pr(X = i) \]

Theorem: Expectation Properties

\[ Ex[c] = c \]

\[ Ex \left[ \sum_{i=1}^{n} X_i \right] = \sum_{i=1}^{n} Ex[X_i] \]
## Theorems: Discrete Random Variables

<table>
<thead>
<tr>
<th>Random variable</th>
<th>HOL Function</th>
<th>Expectation</th>
<th>Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uniform(m)</td>
<td>unif_rv</td>
<td>$m/2$</td>
<td>$\frac{(m+1)^2 - 1}{12}$</td>
</tr>
<tr>
<td>Bernoulli(p)</td>
<td>bern_rv</td>
<td>$p$</td>
<td>$p(1-p)$</td>
</tr>
<tr>
<td>Geometric(p)</td>
<td>geom_rv</td>
<td>$1/p$</td>
<td>$\frac{1-p}{p^2}$</td>
</tr>
</tbody>
</table>
Probabilistic Theorem Proving - Case Studies

- Very few examples
  - Roundoff error analysis of a Digital Processor
    - Verification of a couple of probabilistic properties

- Probabilistic Analysis of Algorithms
  - Miller Rabin Test
  - Coupon-Collector’s Problem
Outline

- Introduction
- Theorem Proving based Performance Analysis
- Performance Analysis of ARQ Protocols
- Conclusions
Automatic Repeat Request (ARQ)

- **Reliable** communication between computers

- **Transmitter**
  - Repeats transmission of a data frame until it receives an ACK

- **Receiver**
  - Discards erroneous data frames
  - Sends Acknowledgment (ACK) for Error-free data frames

- **Applications**
  - Transmission Control Protocol (TCP)
  - High-level Data Link Control (HDLC) Standard
ARQ Protocols

- Implementation variants of ARQ principle
  - Stop-and-Wait
  - Go-Back-N
  - Selective Repeat

- Performance Analysis Metric
  - Message Delay

- Both simulation and state-based formal techniques fail to produce reasonable results
  - A subtle interaction of a number of distributed components
Stop-and-Wait Protocol

Delay (Unsuccessful Transmission Trial)

\[ T_u = t_f + t_{out} \]

Delay (Successful Transmission Trial)

\[ T_s = t_f + t_a + 2(t_{prop} + t_{proc}) \]
Go-Back-N Protocol

- Delay (Unsuccessful Transmission Trial)
  \[ T_u = t_f + t_{out} \]

- Delay (Successful Transmission Trial)
  \[ T_s = t_f \]
Selective Repeat Protocol

**Delay (Unsuccessful Transmission Trial)**

\[ T_u = t_f \]

**Delay (Successful Transmission Trial)**

\[ T_s = t_f \]
Average Message Delay of ARQ Protocols

- $p$: Bit-error probability of the channel
- Average (Message Delay) = ?

Step 1: Message Delay $(T_u,T_s,p)$
- Geometric Random Variable
  - Delay = $(G-1)T_u + T_s$

Step 2: Average of the above random variable
Step 1: Message Delay in HOL

- Geometric random variable function (\texttt{geom_rv})
- Success probability \( = ? \)

- Error behaviour of single bit: \texttt{bern_rv}(p)

**Definition: Frame Error**

\[
\forall n \ p. \quad \text{f_err } 0 \ p = \text{false} \land \\
\text{f_err } (n + 1) \ p = \text{bern_rv}(p) \lor \text{f_err n p}
\]

**Definition: Probability of Successful Transmission**

\[
\forall nf \ na \ p. \quad \text{suc_p_arq } nf \ na \ p = \\
P \{ \text{f_err nf p} \lor \text{f_err na p} = \text{false} \}
\]
Step 1: Message Delay in HOL

Theorem: Probability of Successful Transmission

\[ \forall n_f \, n_a \, p. \quad 0 \leq p \land p \leq 1 \implies s_{uc\_p\_arq} \, n_f \, n_a \, p = (1-p)^{n_f + n_a} \]

Proof

- **Boolean Logic, Positive Integers, Real Numbers, Set, Probability**

Definition: ARQ Message Delay

\[ \forall n_f \, n_a \, p \, T_u \, T_s. \quad arq\_del = T_u \, (geom\_rv \, ((1-p)^{n_f + n_a} - 1) + T_s) \]
Step 2: Average Message Delay

Theorem: Linearity of Expectation

\[ \text{Ex}[aX + b] = a\text{Ex}[X] + b \]

Proof

- Already verified Expectation properties
  \[ \text{Ex}[c] = c \]
  \[ \text{Ex} \left[ \sum_{i=1}^{n} X_i \right] = \sum_{i=1}^{n} \text{Ex}[X_i] \]

- **Boolean Logic, Positive Integers, Real Numbers, Set, Probability**
Definition: Stop-and-Wait Message Delay

\[ \forall nf \ na \ p \ tout \ tprop \ tproc \ tf \ ta. \]
\[ \text{sw}_{-}\text{del} \ nf \ na \ p \ tout \ tprop \ tproc \ tf \ ta = \]
\[ (tf + tout) \ (\text{geom}_{-}rv \ ((1-p)^{(nf + na)}) - 1) \]
\[ + \ tf + ta + 2(tproc + tprop) \]

Theorem: Average Stop-and-Wait Message Delay

\[ \forall nf \ na \ p \ tout \ tprop \ tproc \ tf \ ta. \ (0 \leq p) \land (p < 1) \Rightarrow \]
\[ \text{expec} \ (\text{sw}_{-}\text{del} \ nf \ na \ p \ tout \ tprop \ tproc \ tf \ ta) = \]
\[ (tf + tout) \ (1 - (1-p)^{(nf + na)})/((1-p)^{(nf + na)}) \]
\[ + \ tf + ta + 2(tproc + tprop) \]

Proof:

- \[ Ex[ax + b] = aE[X] + b \]
- Expectation of Geometric random variable
Definition: Go-Back-N Message Delay

\[ \forall \, nf \ na \ p \ tout \ tf. \quad gbn\_del \ nf \ na \ p \ tout \ tf = \]
\[ (tf + tout) \left( \text{geom\_rv} \left( (1-p)^{(nf + na)} \right) - 1 \right) + tf \]

Theorem: Average Go-Back-N Message Delay

\[ \forall \, nf \ na \ p \ tout \ tf. \quad (0 \leq p) \land (p < 1) \Rightarrow \]
\[ \text{expec} \left( gbn\_del \ nf \ na \ p \ tout \ tf \right) = \]
\[ (tf + tout) \left( 1 - (1-p)^{(nf + na)} \right) / \left( (1-p)^{(nf + na)} \right) + tf \]

Proof:
- \[ Ex[aX + b] = aE[X] + b \]
- Expectation of Geometric random variable
Performance Analysis of ARQ Protocols using Theorem Proving

Average Message Delay in HOL
Selective Repeat Protocol

Definition: Stop-and-Wait Message Delay

\[ \forall n_f \ na \ p \ tf. \ sr_{del} \ n_f \ na \ p \ tf = (tf) \ (\text{geom}_r\text{v} \ ((1-p) \ (n_f + na)) - 1) + tf \]

Theorem: Average Stop-and-Wait Message Delay

\[ \forall n_f \ na \ p \ tf. \ (0 \leq p) \land (p < 1) \Rightarrow \text{expec} (sr_{del} \ n_f \ na \ p \ tf) = (tf)/((1-p) \ (n_f + na)) \]

Proof:

- \[ Ex[aX + b] = aE[X] + b \]
- Expectation of Geometric random variable
Outline

- Introduction
- Theorem Proving based Performance Analysis
- Performance Analysis of ARQ Protocols
- Conclusions
Conclusions

- Probabilistic Theorem Proving
  - Model randomness in systems with higher-order-logic random variables
  - Verify probabilistic and statistical properties in a theorem prover
  - Exact Answers
    - Useful for the analysis of Safety critical application

- Performance Analysis of ARQ Protocols
  - Delay Characteristic $\rightarrow$ Higher-order-logic random variable
  - Verification of Linearity of Expectation Property in HOL
    - Results exactly match the paper-and-pencil based analysis methods
      - 100% precise
Conclusions

- Probabilistic Theorem Proving is **not** a “golden solution” to all performance analysis problems
  - Interactive and tedious nature

- **Less critical** sections of the system
  - Simulation

- **Critical** sections of the system that can be **expressed** as a Markov Chain
  - Model Checking

- **Critical** sections of the system that **cannot be handled** by Model Checking
  - Theorem Proving
Thank you

For more information:
http://hvg.ece.concordia.ca

Contact: o_hasan@ece.concordia.ca
Additional Slides
Performance Analysis Basics - Random Variables

- Discrete Random Variables
  - Attain a **countable** number of values
  - Examples
    - Uniform (countable values in an interval \([a,b]\))
    - Bernoulli (True, False)

- Continuous Random Variables
  - Attain an **uncountable** (infinite) number of values
  - Examples
    - Uniform (all real values in an interval \([a,b]\))
    - Exponential (The time between independent events)
Performance Analysis Basics - Properties of Random Variables

- Used to characterize system’s behaviour
  - Probabilistic properties
    - Probability (Multiplier delay = x)
  - Statistical properties
    - Average message delay of a telecommunication protocol
  - Major decision making criteria in performance analysis