Introduction to Network Simulator

Mouhamad Ibrahim and Giovanni Neglia
mibrahim@sophia.inria.fr, gneglia@sophia.inria.fr

www-sop.inria.fr/maestro/personnel/Giovanni.Neglia/ns_course/ns_course.htm

Maestro team

INRIA Sophia-Antipolis - France

Confidence Intervals

- Consider a random variable X, with fixed, but unknow parameter a (e.g. with unknown mean).
- We want to estimate the parameter a looking at samples of the random variable X.
- In particular given $\mathbf{x} = (x_1, x_2, \cdots x_N)$ samples of X, we want to determine an interval $I = [a_l, a_u]$, such that we have a given level of confidence α about the fact that the parameter belongs to the interval ($a \in I$). The interval I is a $\alpha\%$ confidence interval for a.

Confidence Intervals (cont'd)

- How to proceed
  - Given the a-priori density function of a vector of N samples of X ($f(\mathbf{X}|a)$),
  - we consider also a as a random variable and we evaluate through Bayes' theorem the density function $f(a|\mathbf{x}) = f(\mathbf{x}|a)f(a)/f(\mathbf{x})$,
  - intuitively the values for which $f(a|\mathbf{x})$ has higher values are those more *likely* for the parameter a,
  - hence we collect in the interval I the values maximizing $f(a|\mathbf{x})$.
  - $f(\mathbf{x})$ is a constant, and under ignorance prior (i.e. $f(a)$ costant), $f(a|\mathbf{x}) \propto f(\mathbf{x}|a)$, hence we can just consider a function proportional to $f(\mathbf{x}|a)$ (called the *likelihood* function).

Confidence Intervals (cont'd)

- An example: $x \sim \mathbf{N}(\mu, \sigma^2)$, μ is unknown
  - $\overline{x} = \sum_{i=1}^{N} \frac{x_i}{N}$ is distributed as $N(\mu, \sigma^2/N)$, i.e. with density $f(\overline{x}|\mu) = \frac{1}{\sqrt{2\pi\sigma^2/N}} e^{-\frac{(\overline{x}-\mu)^2 N}{2\sigma^2}}$
  - we can consider the likelihood function $L(\mu|\overline{x}) = e^{-\frac{(\overline{x}-\mu)^2 N}{2\sigma^2}}$
  - and determine the interval $I = [\mu_1, \mu_2]$ including the values maximizing $L(\mu|\overline{x})$ up to collect $\alpha\%$ of the area underneath the curve,
  - in this way we get the interval $[\overline{x} - \frac{\sigma}{\sqrt{N}} z_{1-\alpha/2}, \overline{x} + \frac{\sigma}{\sqrt{N}} z_{1-\alpha/2}]$, where $z_{1-\alpha/2}$ is the $1-\alpha/2$ percentile of the standard gaussian distribution, i.e. Prob$(z < z_{1-\alpha/2} = 1 - \alpha/2)$

Confidence Intervals (cont'd)

- An example: $x \sim \mathbf{N}(\mu, \sigma^2)$, μ and σ^2 unknown
  - the interval is $[\overline{x} - \frac{\hat{\sigma}}{\sqrt{N}} t_{1-\alpha/2}, \overline{x} + \frac{\hat{\sigma}}{\sqrt{N}} t_{1-\alpha/2}]$, where $\hat{\sigma}^2 = \sum_{i=1}^{N} (x_i - \overline{x})^2 / (N-1)$ is an estimator of the variance and $t_{1-\alpha/2}$ is the $1 - \alpha/2$ percentile of the student t distribution with $N - 1$ degrees of freedom.
- with other distributions
  - the central limit is often invoked to apply above expressions for the evaluation of confidence intervals for the mean.

First Assignment

- Starting from the network scenario with two CBR sources considered in the previous lessons (`cbr.tcl`), let the two sources start uniformly at random in the interval $[0.5, 1.5]$. Develop a script (a set of scripts), in order to perform different runs and evaluate the confidence intervals for the average throughput of the two sources. Remarks/Suggestions:
  - Use different generators for independent random variables and a different substream for a given generator at each run.
  - Use the agent LossMonitor as sink `set sink0 [new Agent/LossMonitor]` in order to be able to read the data received through the variable `bytes_` of the agent.
  - Evaluate the average throughputs at the end of the ns simulations and append them to a file, then calculate confidence intervals through post processing.

Second Assignment

- In the above network scenario change the rate of one CBR source to 0.5 Mbps and replace the other one with a FTP source, which uses TCP with MSS equal to 552 bytes. The advertised window from the receiver is equal to 8000 packets. Introduce a procedure that logs the congestion window value every 10 ms. Write a script to evaluate the average throughput of the TCP source every 10 ms from the global ns trace file (the one obtained by `trace-all`). Visualize the window and the throughput time evolution.

Some useful tcl commands

- Create a TCP NewReno source

```
set tcpSource [new Agent/TCP/Newreno]
```

- Create a TCP sink with delayed acks

```
set sink [new Agent/TCPSink/DelAck]
```

- Access the congestion window value (packets #)

```
$tcpSource set cwnd_
```

- Access the receiver window value (packets #)

```
$tcpSource set window_
```

In ns the receiver window is set to a constant value at the sender

- Setup a FTP source over a TCP connection

```
set ftp [new Application/FTP]
$ftp attach-agent $tcpSource
```

How to plot

- gnuplot, xgraph, matlab, octave, ...
  - e.g. with gnuplot one can plot data in the second column versus data in the first column with

    ```
    plot 'nomefile' using ($1):($2)
    ```