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

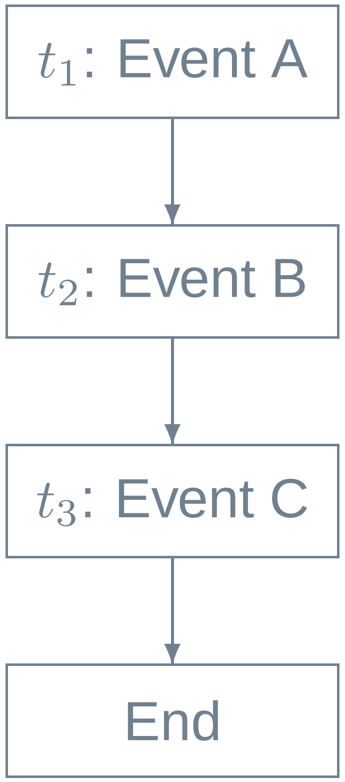
Maestro team

INRIA Sophia-Antipolis - France

About Simulations

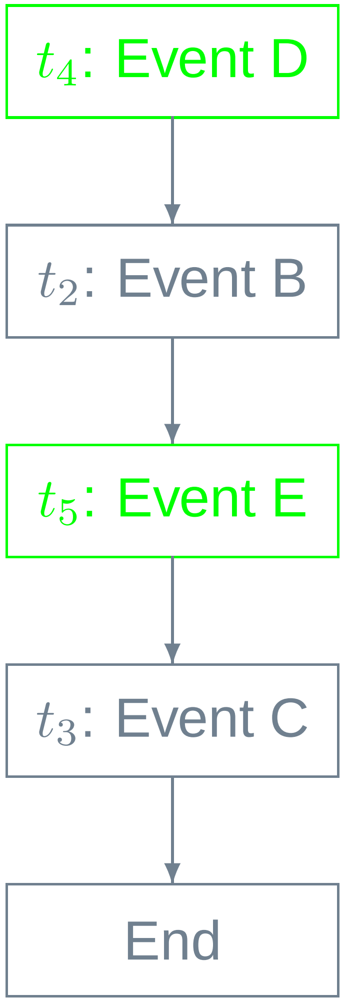
- “The technique of imitating the behaviour of some situation or system (Economic, Mechanical etc.) by means of an analogous model, situation, or apparatus, either to gain information more conveniently or to train personnel” (Oxford English Dictionary).
- Approaches to simulations
  - Discrete time systems models.
  - Continuous systems models.
  - Discrete events models.
- NS is a discrete events simulator
  - Calendar or Scheduler.

Calendar



- $t_1 \leq t_2 \leq t_3$
- The simulator executes the event on top of the list.
- The execution of an event can generate other events that are placed in the list according to their time schedule.
- The simulation ends when the event *End* is reached.

Calendar



- $t_4 < t_2 \leq t_5 \leq t_3$
- The simulator executes the event on top of the list.
- The execution of an event can generate other events that are placed in the list according to their time schedule.
  - e.g. the execution of event A has generated events D and E, event D is on top of the list (because $t_4 < t_2$) and hence is the next event to be executed.
- The simulation ends when the event *End* is reached.

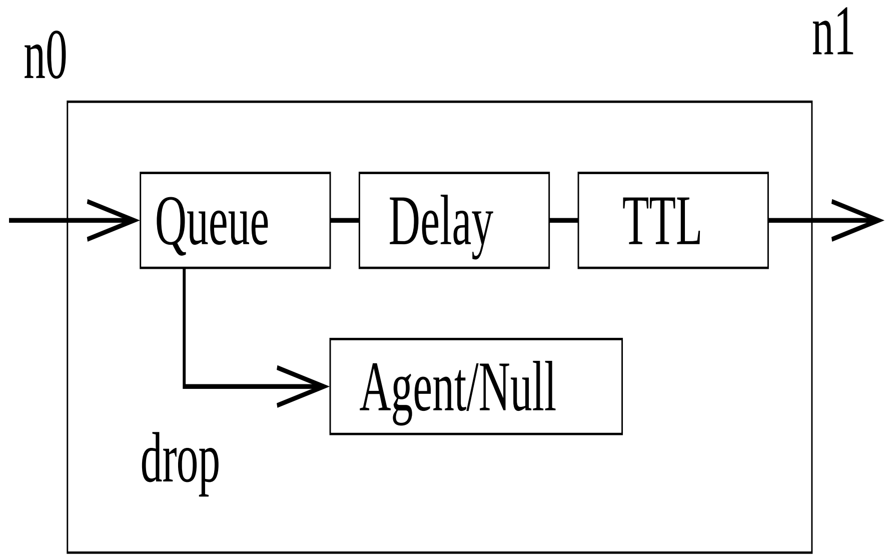
Trace

- nam trace
  - at the begin
    ```
    set nf [open out.nam w]
    $ns namtrace-all $nf
    ```
  - at the end
    ```
    $ns flush-trace
    close $nf
    ```
- events trace
  ```
  set tracefile1 [open out.tr w]
  $ns trace-all $tracefile1
  ```
- tracing a specific object
  - e.g. `$ns trace-queue $n2 $n3 $file1`

Trace (inside NS)

- NS internal structure of a link:

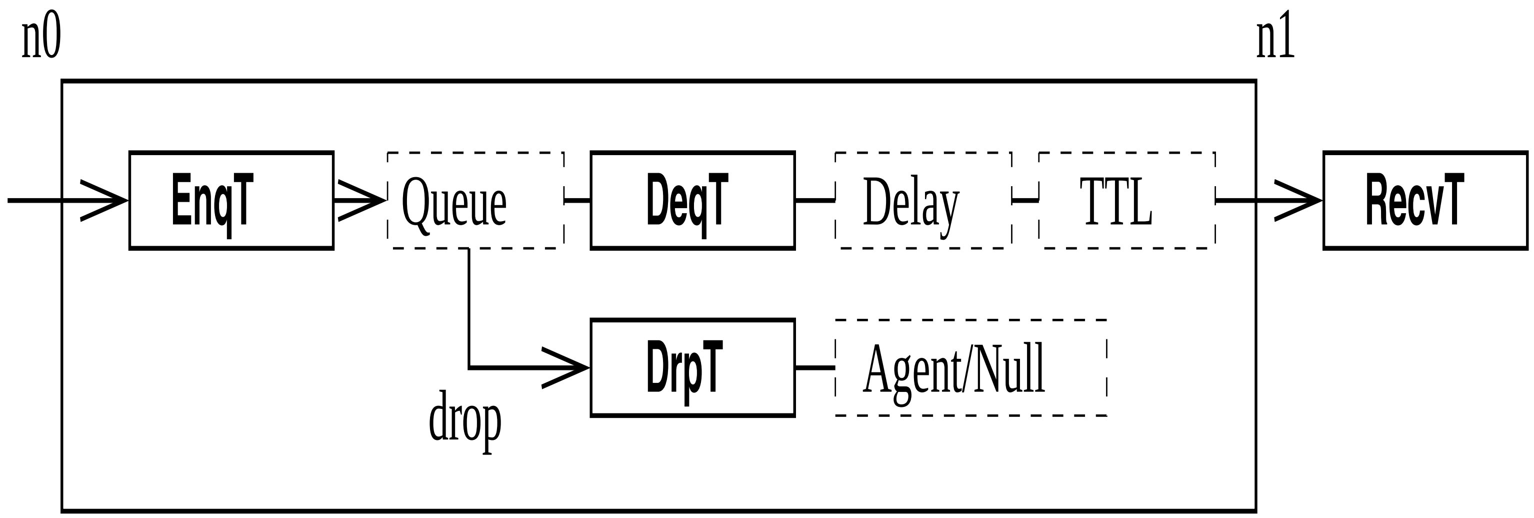
Figure from *NS Simulator for beginners, E. Altman and T. Jiménez*



Trace (inside NS)

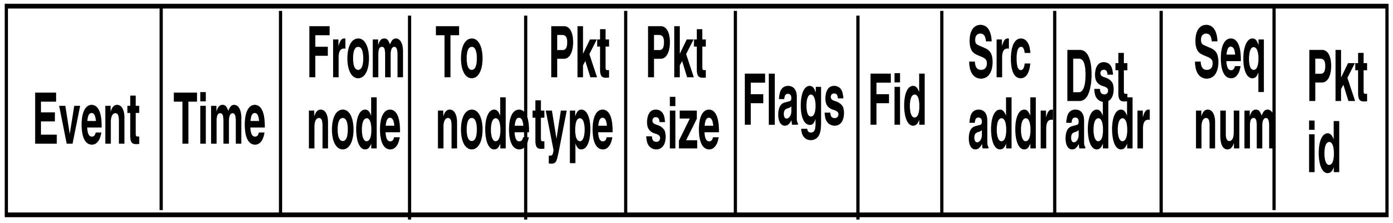
- NS internal structure of a traced link:

Figure from *NS Simulator for beginners,* *E. Altman and T. Jiménez*



Trace file format

Figure from *NS Simulator for beginners*, *E. Altman and T. Jiménez*

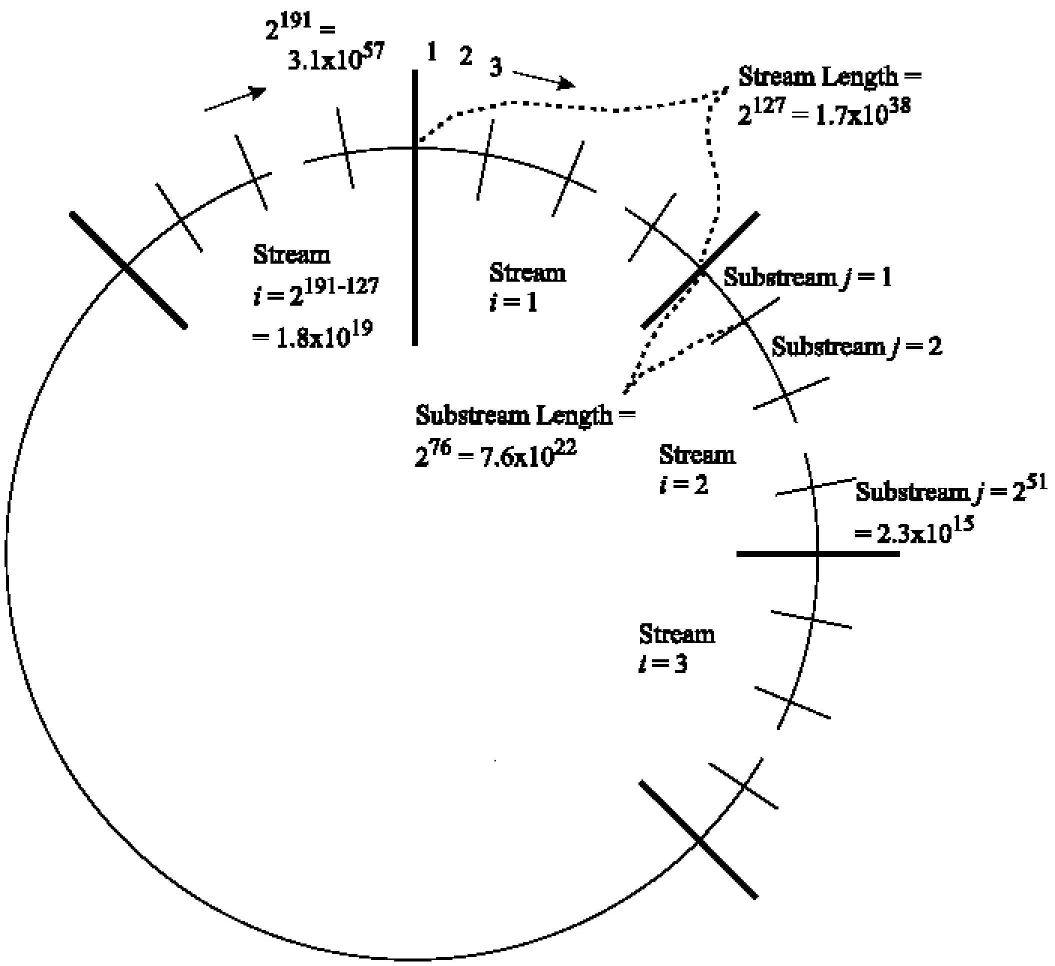


Event	Time	From node	To node	Pkt type	Pkt size	Flags	Fid	Src addr	Dst addr	Seq num	Pkt id

Random Variables (RVs)

- How can a machine generate a random number?
- Random number generators in NS
  - combined multiple recursive generator MRG32k3a
  - `set Rng1 [new RNG]`
- Seeds `$Rng1 seed $a`
  - raw
  - pre-defined (independent streams divided in substreams),
  - heuristic seeds `$Rng1 seed 0`

Streams and substreams



RVs from random generators

- Cumulative Distribution Function method
- Predefined random variables, e.g.

```
set r1 [new RandomVariable/Uniform]
$r1 use-rng $Rng1
$r1 set min_ 0.0
$r1 set max_ 10.0
```

- How to use
  - a predefined seed for each independent variable
  - a different substream for each independent run

```
$r1 next-substream
```

When are RVs needed

- as a black-box model of aspects we prefer to ignore
- when the process to simulate is intrinsecally stochastic
- when we want to average over different unknown initial conditions
  - but we could also look at steady state