



|| at,0|a
One at,0 20
or v0 at,0
@ at,0|00
sc at,0|01)









Experiences with the ns-2 Network Simulator

Explicitly Setting Seeds Considered Harmful



Martina Umlauf
umlauft@wit.tuwien.ac.at

Peter Reichl
reichl@ftw.at



Overview

- The ns-2 RNG
- Examples:
 - Correlation Experiment
 - Wired Topology Example
 - Wireless Example
- Impact on Real-world Simulations
- How to Avoid

The ns-2 RNG

- ns-2 versions $\leq 2.1b8$: **old RNG**
Minimal Standard multiplicative Linear Congruential Generator [Park and Miller, 1988]
Period $p=2^{31}-2$
API: `$rng seed $s;`
Sensitive to seeds [Entacher, Hechenleithner, 2002]
- ns-2 versions $> 2.1b9$ until today: **new RNG**
MRG32k3a [L'Ecuyer, 1999]
Period $p = 3.1 \times 10^{57}$
API: `$rng next-substream;`
Promises to fix seed-sensitivity

Correlation Experiment



- Set up 3 RNGs – old method:


```
for {set i 0} {$i < 3} {incr i} {
  set rng($i) [new RNG]
  $rng($i) seed $n($i)
  set u($i) [new RandomVariable/Uniform]
  $u($i) use-rng $rng($i)
}
```

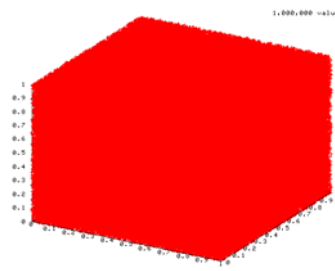
Seed set defined before
- Draw triplets of values $\$u(\$i)$ from $\$rng(\$i)$
- Interpret as vector $\langle \$u(1), \$u(2), \$u(3) \rangle$
- Sets of Seeds:

1973272912	1	1
1822174485	2	634005912
1998078925	3	634005912

Set 1
good
Set 2
bad
Set 3
bad






Expected Result for Uniform Values

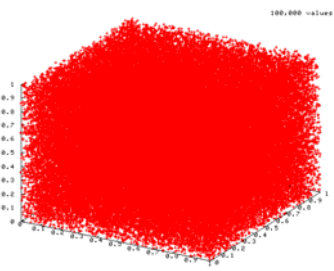


1,000,000 values

1,000,000 triplets drawn





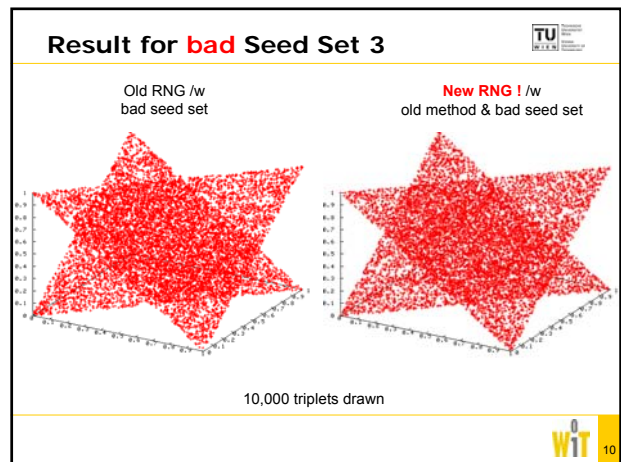
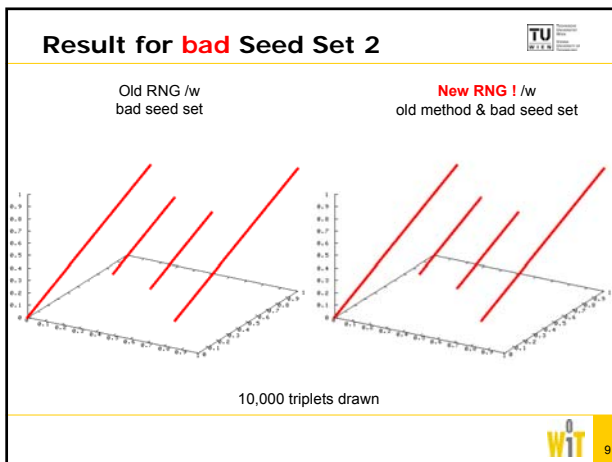
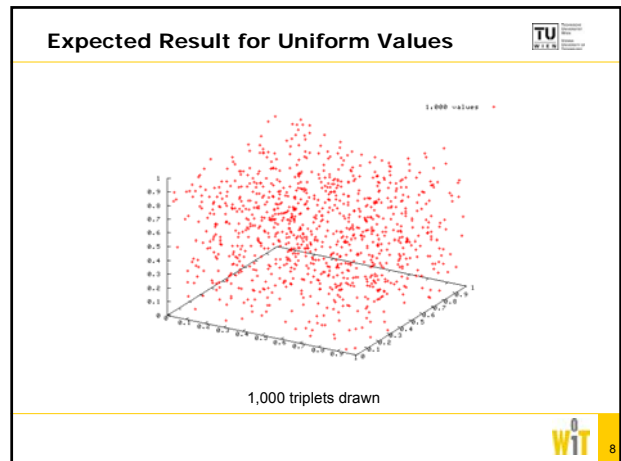
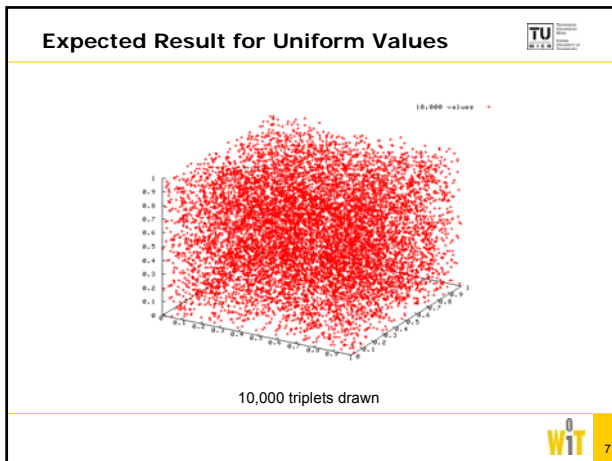
Expected Result for Uniform Values



100,000 values

100,000 triplets drawn





Wired Topology Example / 1

Exponential On/Off-Traffic:
 On: 0.08μs / 1 pkt / 1000bytes
 Off: 41ms

$\lambda = 8000\text{bits} / 41\text{ms}$
 $= 0.195\text{Mbps}$
 $\Sigma\lambda = 0.976\text{Mbps}$
 $\rho = \Sigma\lambda / \text{BW} = 0.976 \dots \text{utilization factor}$

Expected mean queue length:

$$\bar{q} = \frac{\rho}{1-\rho} - \frac{\rho^2}{2(1-\rho)} = 20.488\text{pkts}$$

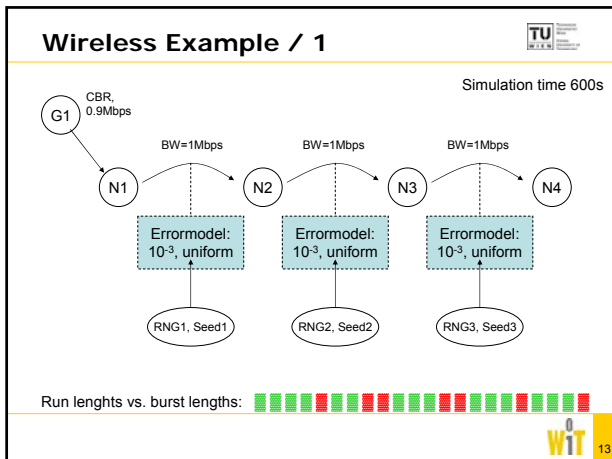
Simulation time 7200s
 Sampling time 10ms

Wired Topology Example / 2

	Seed Set 1 good	Seed Set 2 bad
1973272912	1	
1822174485	2	
1998078925	3	
678622600	4	
999157082	5	

New RNG	New method	20.2996
New RNG	Set 2 - bad	29.4527
Old RNG	Set 1 - good	19.4398
Old RNG	Set 2 - bad	24.2785

Expected: 20.488



Wireless Example / 2

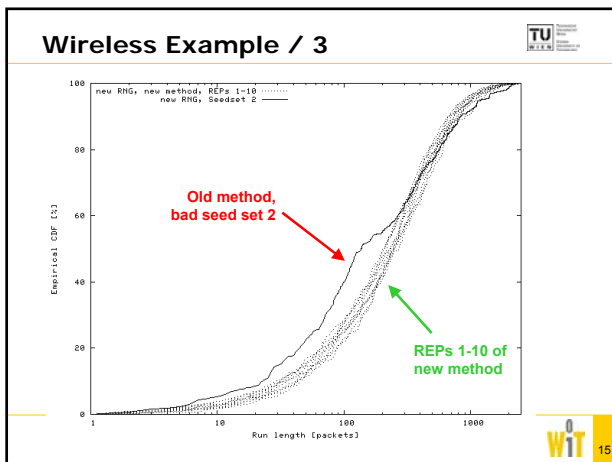
- New RNG / old method / bad seed


```
$rng1 seed 1;
$rng2 seed 2;
$rng3 seed 3;
```

vs.

- New RNG / new method


```
for {set i 1} {$i < $rep} {incr i} {
  $rng1 next-substream;
  $rng2 next-substream;
  $rng3 next-substream;
}
```



Impact on Real-world Simulations

Postings on ns-users mailing list (2005, 2006)

Advice or example incorrectly using old method	22
Correct advice in response to seeding question	2
Example containing correct method in other context	7
Ambiguous example or advice	4
Advice to use consecutively numbered seeds	2

28 incorrect vs. 9 correct !

Results affected if

- Seeds are chosen badly (eg. 1,2,3,...)
- Several RNG objects used

How to Avoid

- Do NOT use:


```
$rng seed <value> where <value> <= REP
```
- Only USE:


```
for {set i 1} {$i < $rep} {incr i} {
  $rng1 next-substream;
  $rng2 next-substream;
  $rng3 next-substream;
  ...
}
```
- Optionally:


```
global defaultRNG;
$defaultRNG seed <value>
```

Thank You!