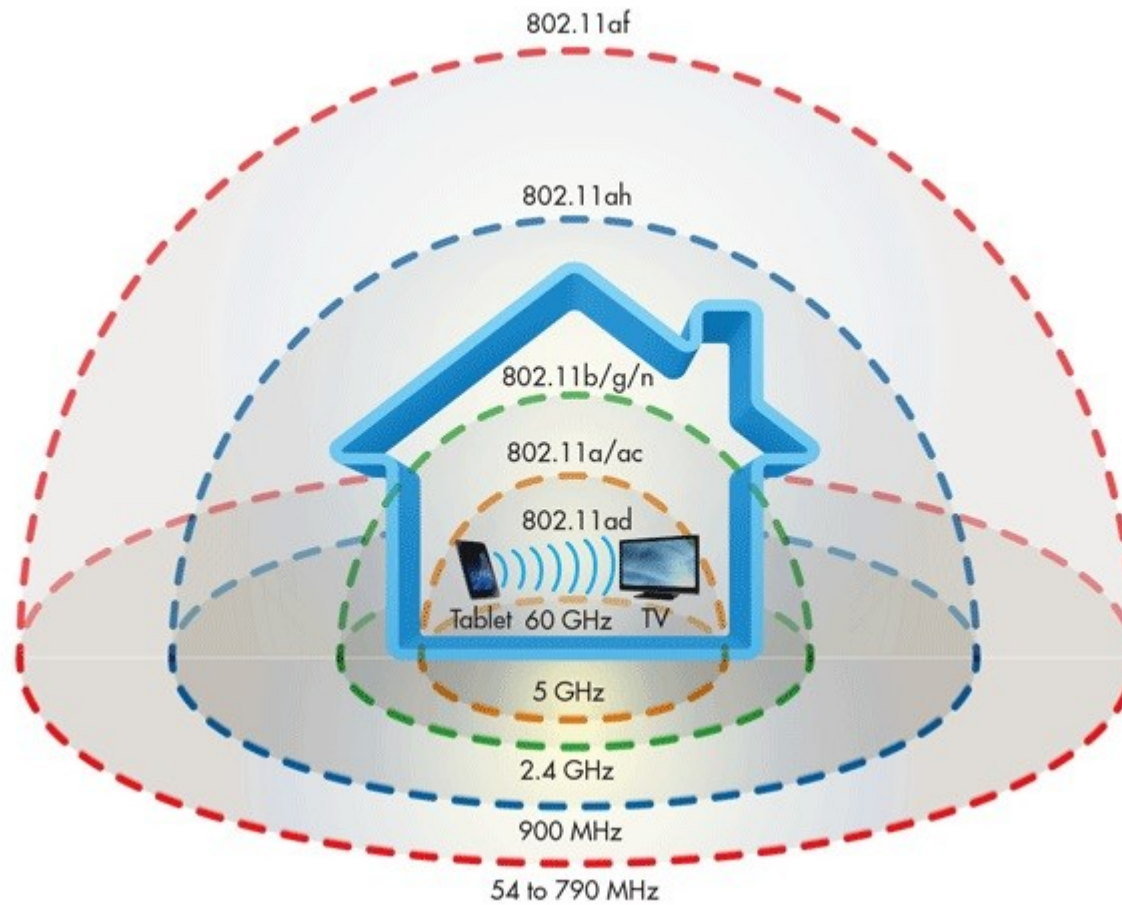


Počítačové siete 2

Modely pre generovanie paketov, simulačné
nástroje

Martin Drozda

802.11



Modely pre generovanie paketov

CBR = constant bit rate

Generovanie paketov v konštantných intervaloch

Napr. Glomosim:

CBR 947 1174 4000 128 0.025S 0S 100S

VBR = variable bit rate

Generovanie paketov podľa daného rozdelenie (ale nie uniformne)

Modely pre generovanie paketov

Pre ns3:

```
OnOffHelper onOffHelper ("ns3::UdpSocketFactory",  
InetSocketAddress (Ipv4Address ("10.0.0.2"), cbrPort));
```

```
onOffHelper.SetAttribute ("PacketSize", UIntegerValue  
(1400));
```

```
onOffHelper.SetAttribute ("OnTime", StringValue  
("ns3::ConstantRandomVariable[Constant=1]"));
```

```
onOffHelper.SetAttribute ("OffTime", StringValue  
("ns3::ConstantRandomVariable[Constant=0]"));
```

```
onOffHelper.SetAttribute ("DataRate", StringValue  
("3000000bps"));
```

```
onOffHelper.SetAttribute ("StartTime", TimeValue (Seconds  
(1.000000))));
```

```
cbrApps.Add (onOffHelper.Install (nodes.Get (1)));
```

Poissonov model:

$$P(N_t = k) = e^{-\lambda t} \frac{(\lambda t)^k}{k!}$$

Stredná hodnota = λ , rozptyl = λ , kde λ je parameter.

Pre $k = 0, 1, 2, 3, \dots$

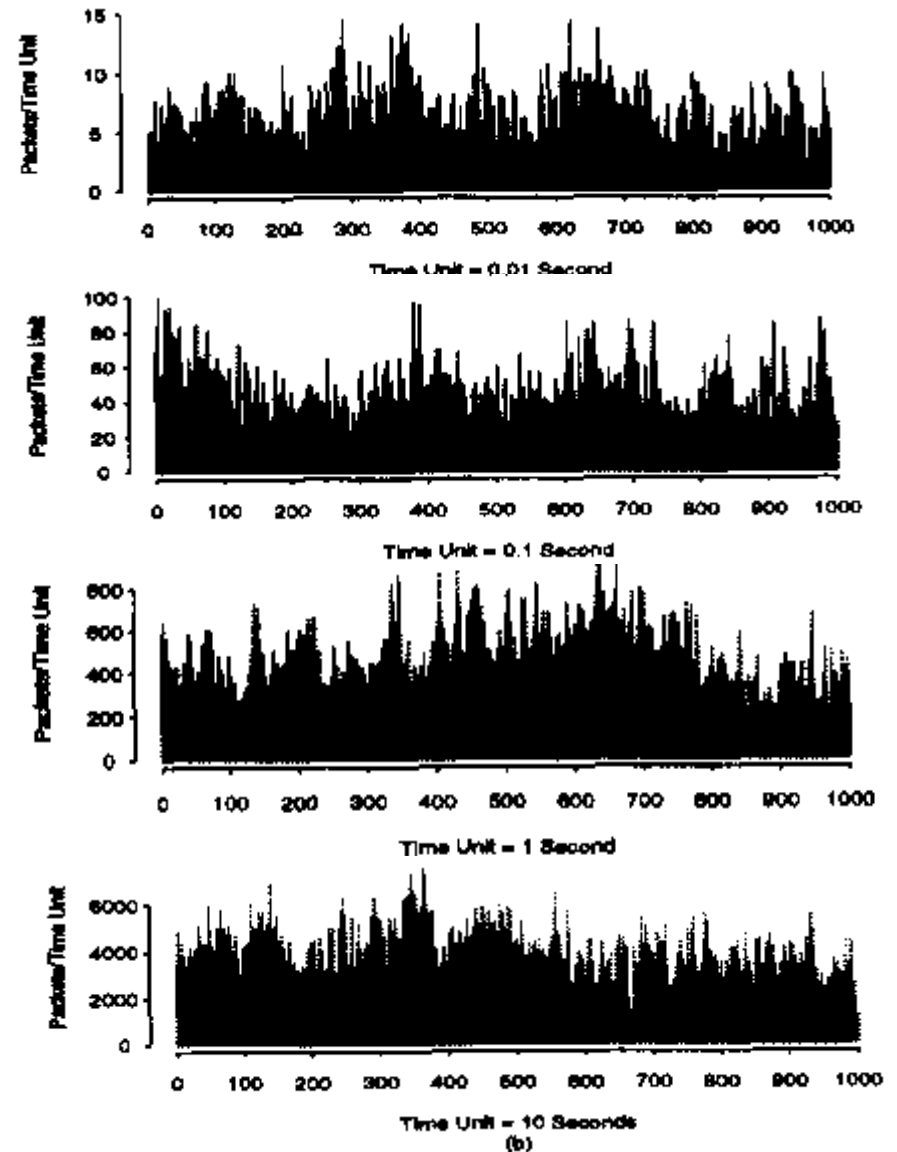
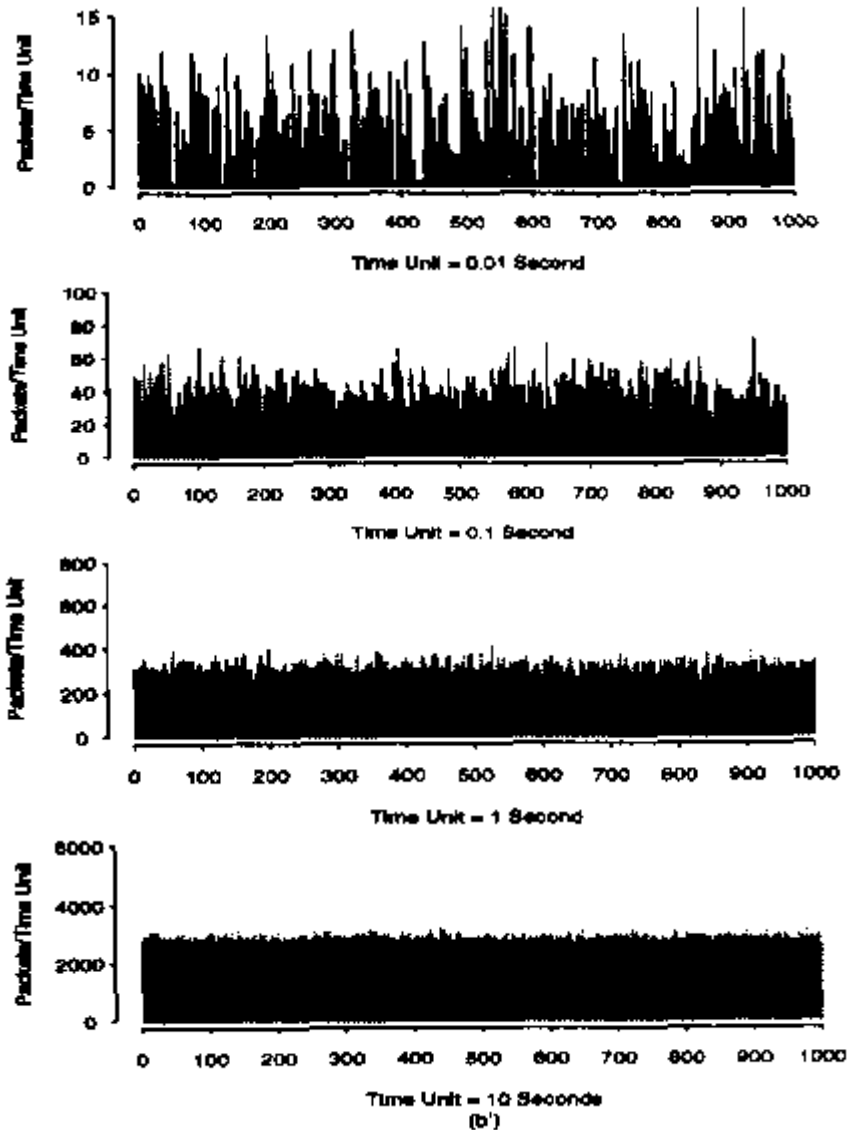
Pravdepodobnosť výskytu udalosti do času $t = k$.

$P(N_t = 0) = e^{-\lambda t}$ Pravdepodobnosť výskytu prvej udalosti.

Poisson

vs

realita



Experiment

Dáta z Ethernetovej siete z rokov *1989-1992*.

DEC 8650 minicomputers, SPARCstation 1...

Aplikácie sa zmenili, správanie používateľov sa zmenilo.

Poisson: čas udalostí je nezávislý

Reality: udalosti sú závislé (korelované)

$$r(k) = \frac{E[(x_t - \bar{x})(x_{t+k} - \bar{x})]}{s^2} ; \text{ autokorelačná funkcia}$$

$$r(0) = \frac{E[(x_t - \bar{x})(x_t - \bar{x})]}{s^2} = 1$$

ak x_t, x_{t+k} sú identicky rozdelené a nezávislé potom platí:

$$r(k) = \frac{E[(x_t - \bar{x})(x_{t+k} - \bar{x})]}{s^2} = \frac{E[(x_t - \bar{x})]E[(x_{t+k} - \bar{x})]}{s^2} = 0$$

Autokorelačná funkcia má v mnohých prípadoch tvar:

$$r(k) \sim k^{-b}$$

$$H = 1 - b/2$$

b má súvis s Hurstovým parametrom, ktorý je mierou dlhodobej závislosti v časovej sérii

Mnoho systémov má Hurstov parameter rovný ~ 0.74

Brownov pohyb má $H = 0.5$

Dôsledok: pakety sú generované v nárazoch a dôvod môže byť časovo vzdialený

Simulačné nástroje

Glomosim (UCLA)– Qualnet: Parsec (podobné jazyku C)

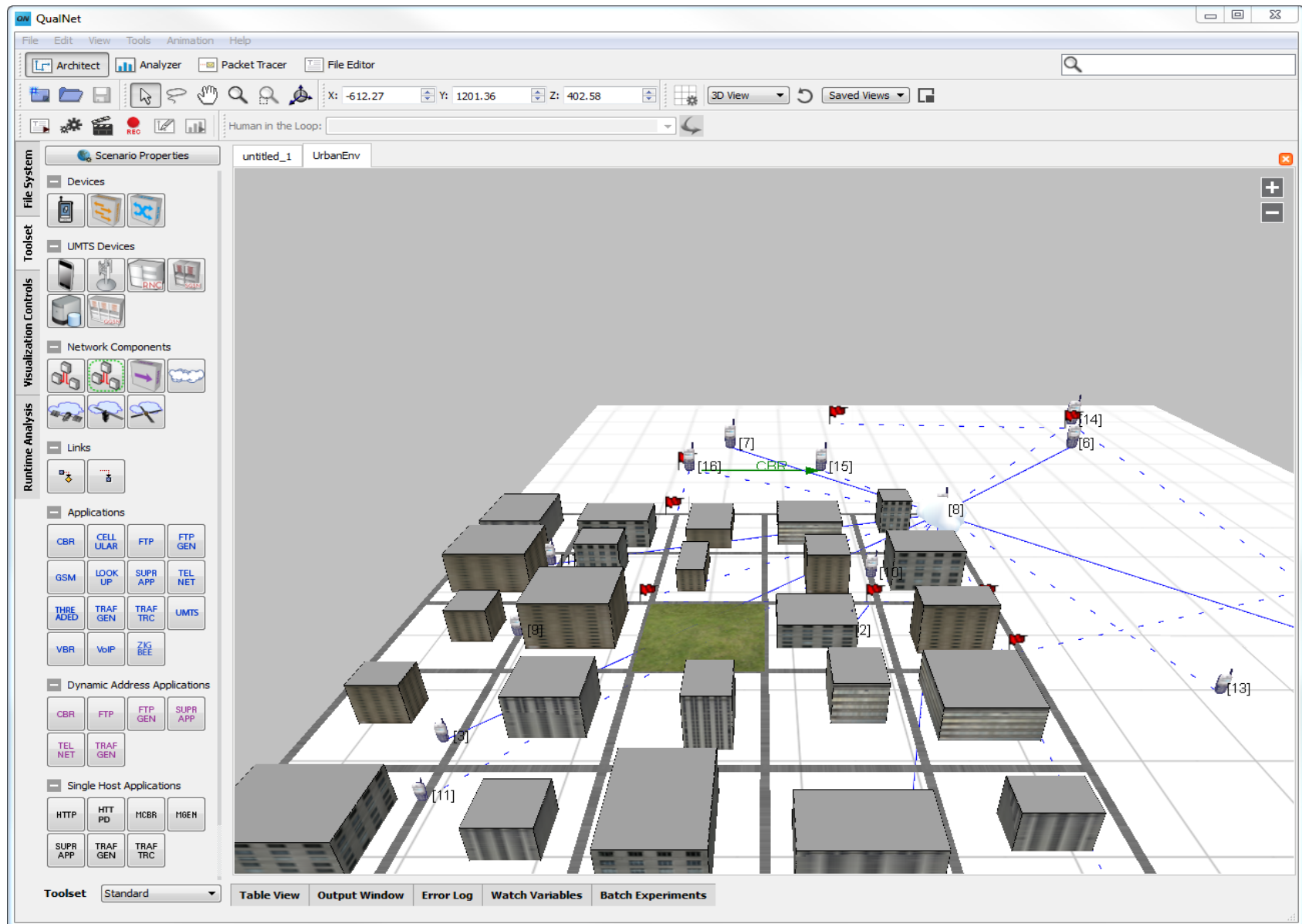
ns2 (USC): C++/OTcl

ns3: C++

JiST/swans (Cornell): Java

OPNET

OMNet++



Simulácia je...

Technika imitácie správania sa systémov pomocou analogických modelov, situácii a aparátu s cieľom získať informáciu alebo trénovať ľudí.

"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."

(from the Oxford dictionary)

- *Spojité systémy*

Zmena nastáva spojite

- *Diskrétné systémy*

Zmena nastáva v dopredu určených okamžikoch

- *Udalostné systémy*

Zmena nastáva po (náhodných) udalostiach...a po zmenách stavu

- V pravidelných časových intervaloch $t_0, t_1, t_2, \dots$ (synchronná simulácia). Nevýhoda: intervale môžu byť malé.
- Po udalostiach (asynchronná simulácia), každá udalosť môže spôsobiť zmenu stavu

- *System*

Množina objektov, ktoré navzájom interagujú (napr. bezdrôtová sieť).

- *Model*

Matematická reprezentácia systému

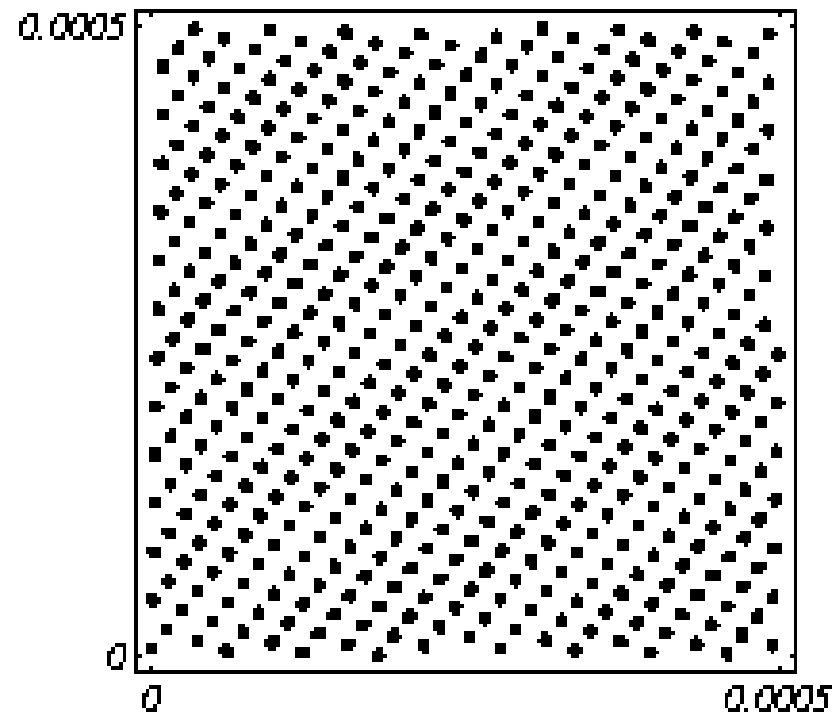
- *Entita*

Objekt systému napr. uzol siete.

- *Atribút*

Vlastnosť entity, napr. počet generovaných paketov.

Moja simulácia je taká dobrá ako...

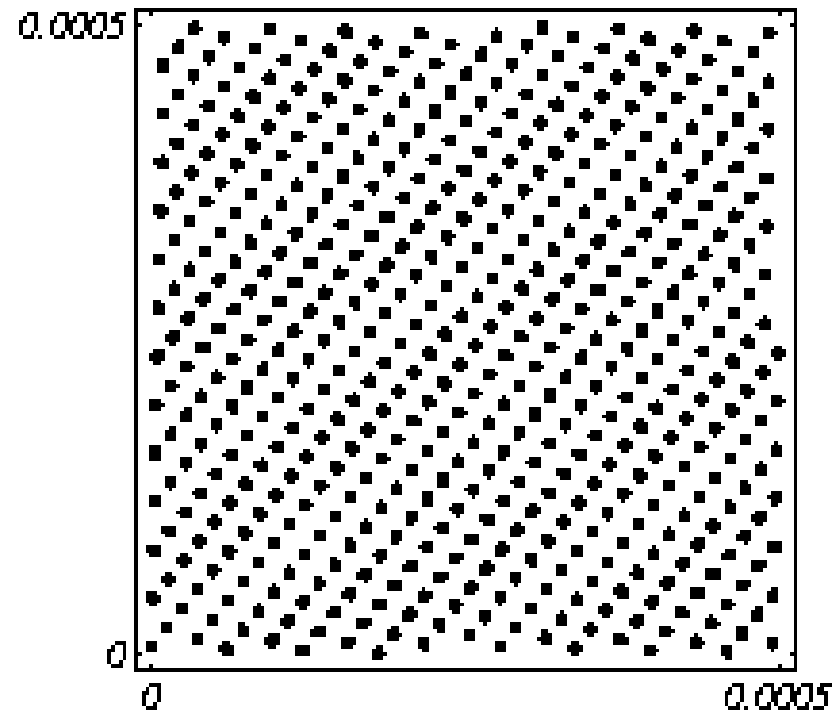


Linear congruential generator (LCG)

$$y_{n+1} = a y_n + b \pmod{m}$$

ANSI C: `rand()`

$a = 1103515245$
 $y_0 = 12345$ (*SEED*)
 $b = 12345$
 $m = 2^{31}$



Použi:

SeedManager::SetSeed (3);

alebo

SeedManager::SetRun (7);

Vlastnosti:

- *Periódá: 10^{57}*
- *Vlastnosti analyzované teoreticky*
- Podpora vláken

ns3::UniformVariable

ns3::ExponentialVariable

*ExponentialVariable x(3.14);
x.GetValue(); //will always return with mean 3.14*

ns3::NormalVariable

ns3::WeibullVariable

Kvalita vs rýchlosť

Kvalita:

- Simuluje všetky vrstvy OSI ref. modelu?
- Sú protokoly zjednodušené?

Čas a priestor:

- Rýchlosť: reálny čas
- Priestor: škálovanie s počtom simulovaných uzlov

Lineárne hľadanie ($O(n)$):

Nájdí všetkých susedov a zisti, či sú dosiahnuteľní.

Flat binning:

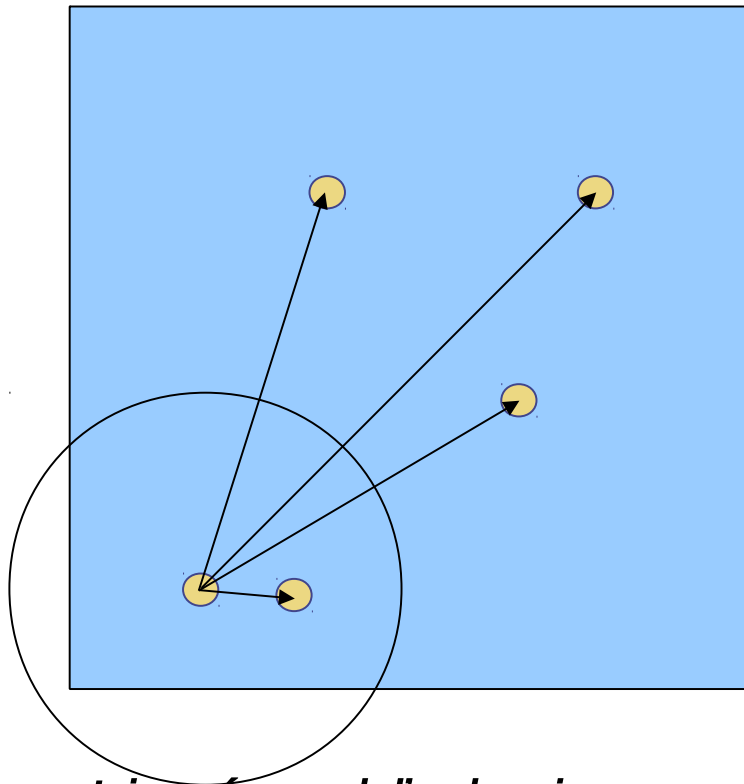
Simulovaná oblasť je pravidelne rozdelená na štvorce. Hľadáj vo štvorcoch, ktoré sú do určitej vzdialenosti. Vyžaduje vedomosť o príslušenstve k štvorcu.

Hierarchické hľadanie:

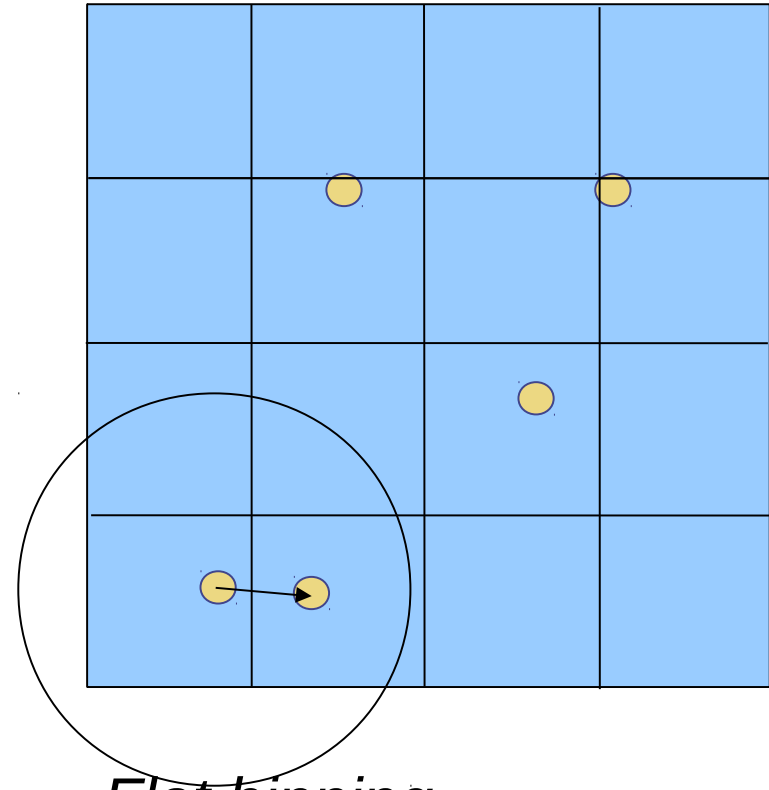
Rekurzívne rozdel' simulovanú oblasť pozdĺž osí x a y

Predpoklady: uniformne rozdelené uzly (v jednom štvorci relatívne nízky počet uzlov), akceptovateľná zmena polohy uzlov

Dátové štruktúry pre šírenie signálu

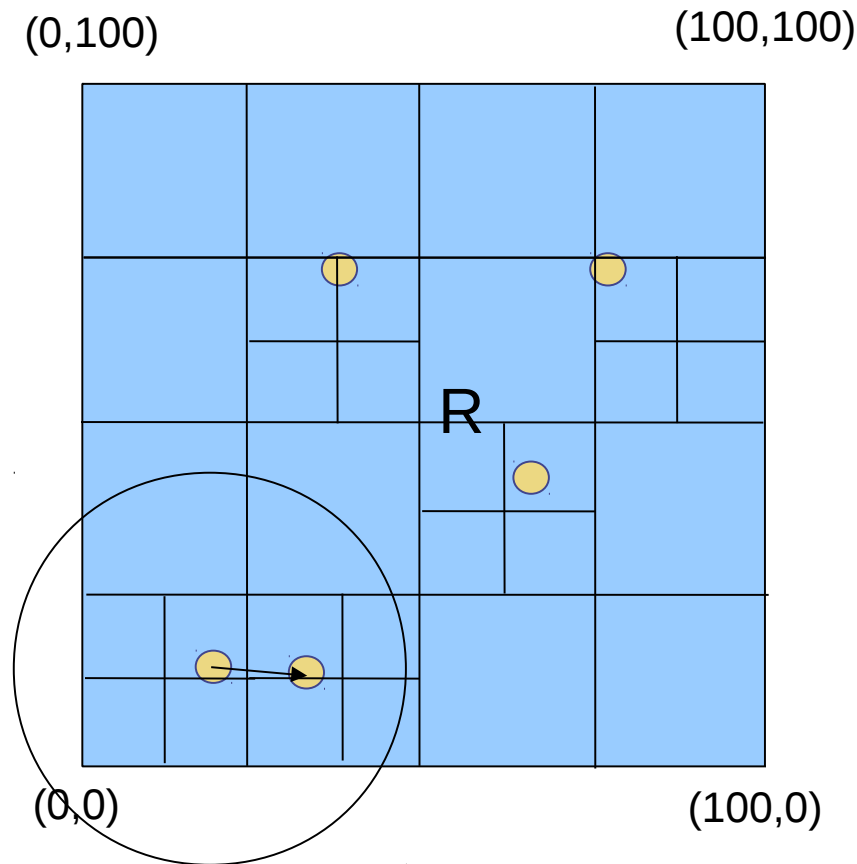


Lineárne hľadanie
Vzdialenosť počítaná
pre každý uzol



Flat binning
Vzdialenosť počítaná
pre uzly v susedných
štvorcoch. Príslušnosť
uzla ku štvorcu je
potrebná.

Hierarchické hľadanie



Priestorový dekompozičný strom. Deliace body sú fixné koordináty. Štvorce sú listy.

$$height = \log_4(field_size/bin_size)$$

Príslušnosť ku štvorcu je potrebná.

Začneme v koreni R a delíme priestor rekurzívne. Podobné ako binárne hľadanie, ale v 2D. Je stred štvorca v dosahu signálu?

```
17      3      14399.047414720 14399.098528933 145      420
    127      420      600      572      1128.00 1815.00 0.00      1718
59      -1      14408
```

17 = protocol type (UDP)
3 = 802.11 packet type (DATA)
Time packet dequeued
Simulation time
145 = MAC from
420 = MAC to
127 = IP from
420 = IP to
Packet size 1
Packet size 2

Glomosim: výstup

```
SIMULATION-TIME  100S
SEED              0
TERRAIN-DIMENSIONS (700, 700)
NUMBER-OF-NODES   49

#NODE-PLACEMENT   FILE
#NODE-PLACEMENT-FILE ./pos.in
NODE-PLACEMENT     GRID
GRID-UNIT          100
# NODE-PLACEMENT   RANDOM
#NODE-PLACEMENT    UNIFORM
```

Ostatné parametre

ns-3 is a free, open source software project building and maintaining a discrete-event network simulator for research and education

Simulácie sú implementované v C++

Používa Doxygen

ns-3 má GNU GPLv2 licenciu

Používa *waf*: konfiguračný framework používajúci Python

<http://code.google.com/p/waf/>

Žiadna spätná kompatibilita s ns2 (ns2 používal Tcl pre implementáciu simulácií)

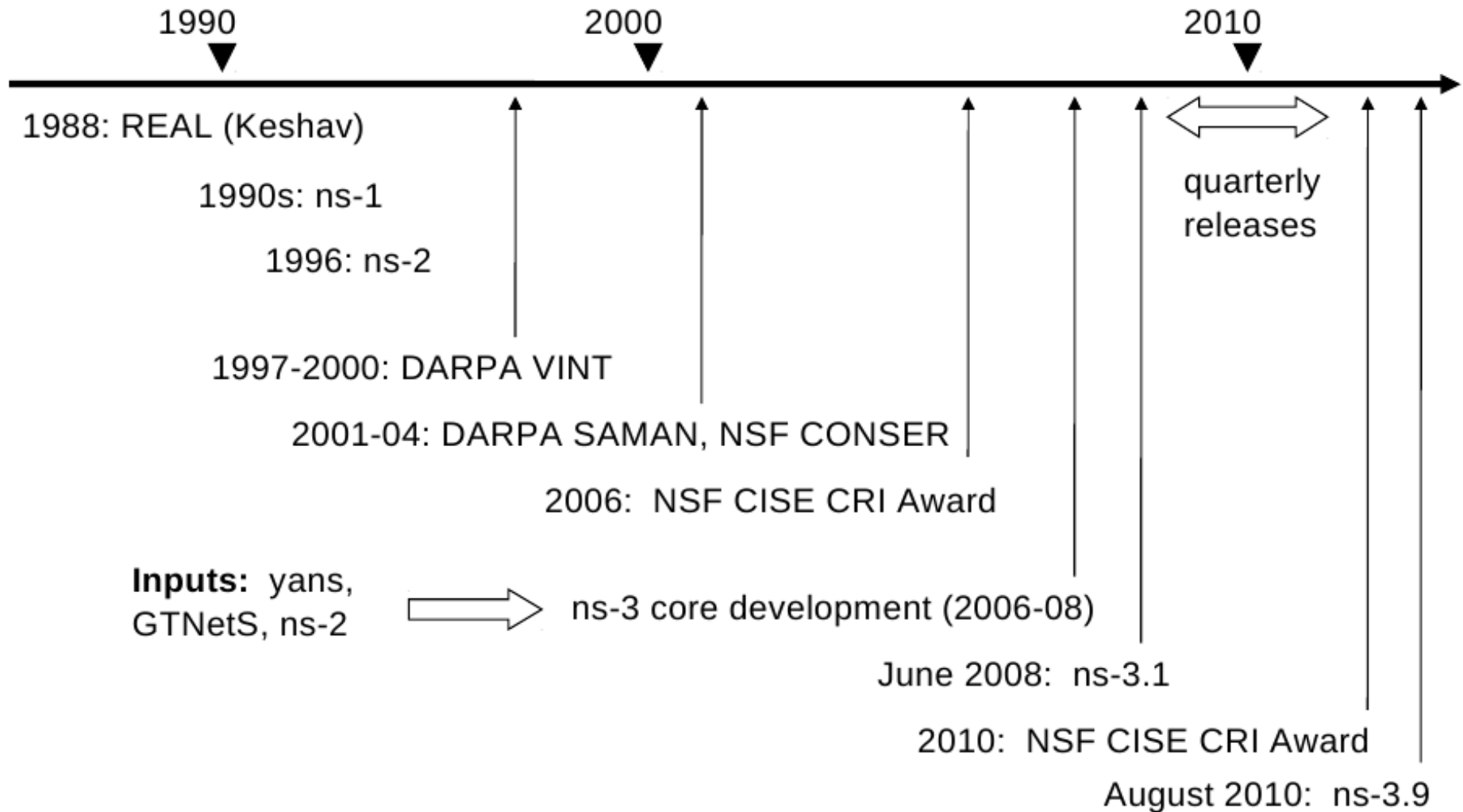


Figure source: ns-3 tutorial at 9th GENI Engineering Conference (GEC9), 2010

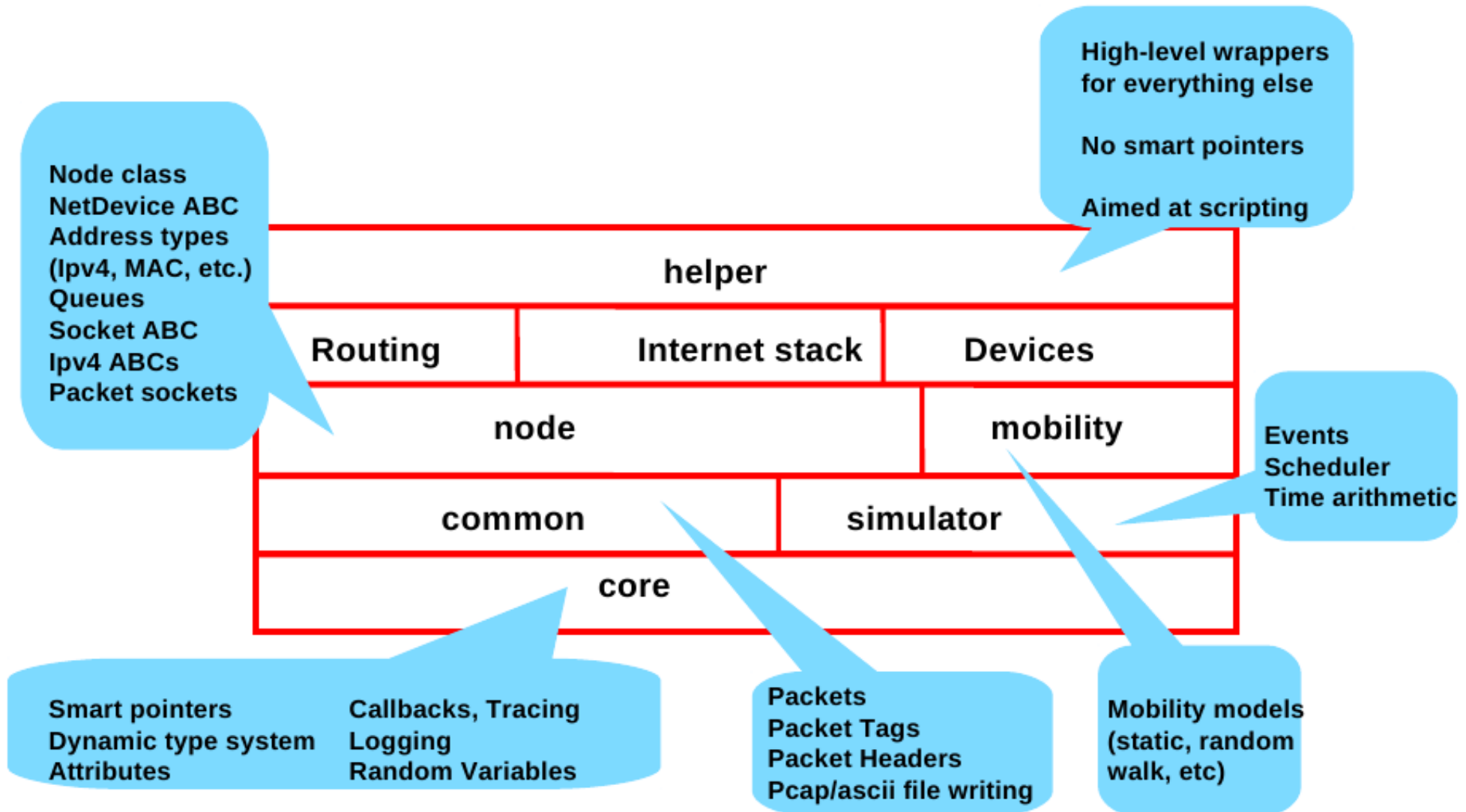


Figure source: ns-3 tutorial at 9th GENI Engineering Conference (GEC9), 2010

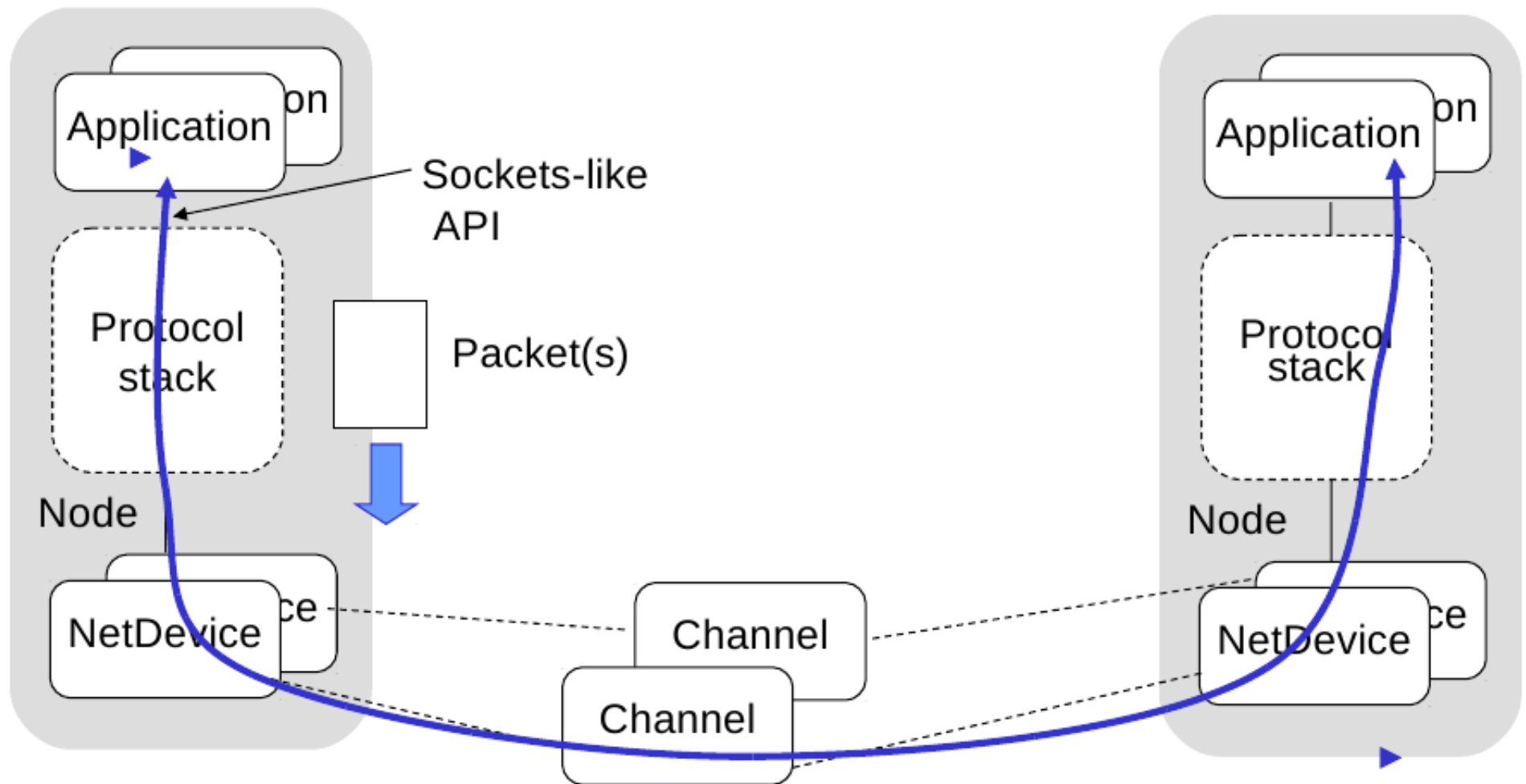


Figure source: ns-3 tutorial at 9th GENI Engineering Conference (GEC9), 2010

```
int main (int argc, char *argv[])
{
    // Set default attribute values
    // Parse command-line arguments
    // Configure the topology; nodes, channels, devices, mobility
    // Add (Internet) stack to nodes
    // Configure IP addressing and routing
    // Add and configure applications
    // Configure tracing
    // Run simulation
}

„./waf --run scratch/filename“
```

Java in Simulation Time / Scalable Wireless Ad hoc Network Simulator

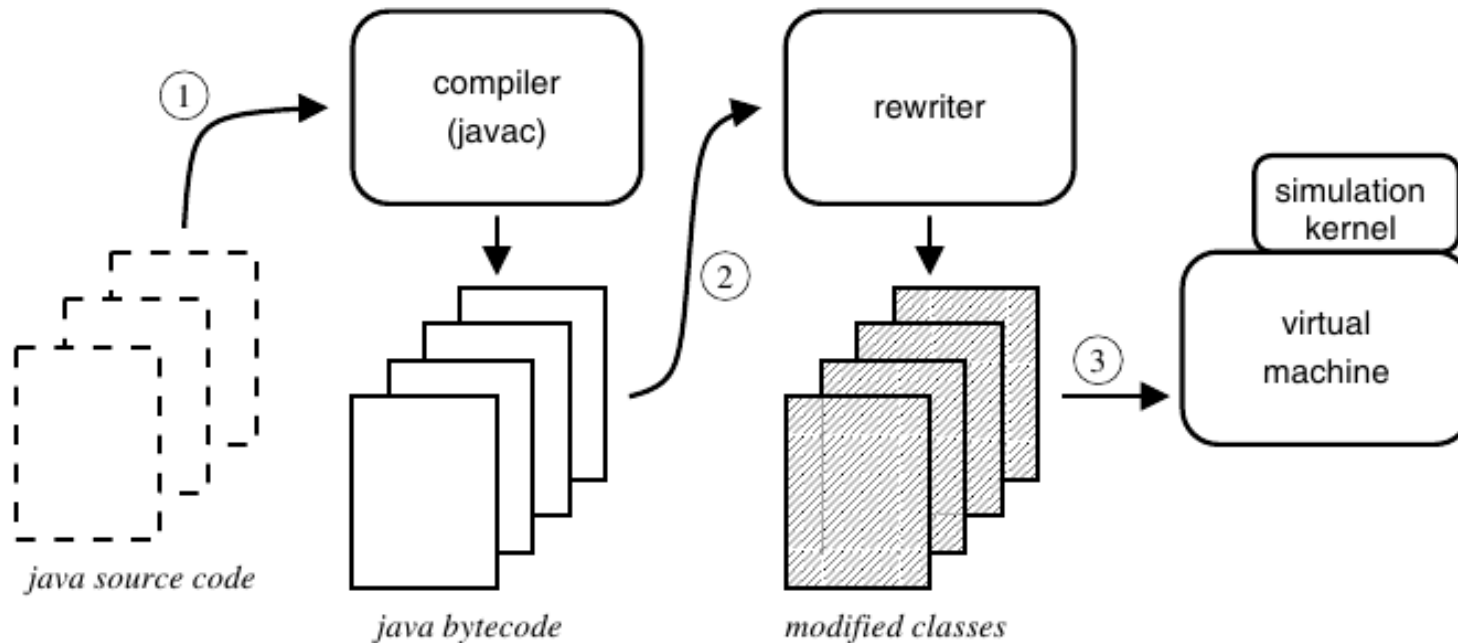
JiST = simulation engine

swans = libraries for ad hoc wireless networks

Vstup: java, jython, bean shell.

Výstup: špecifikovaný používateľom

Výhody: jednoduchá integrovateľnosť s existujúcim kódom, napr. Server implementovaný v Jave môže byť priamo zahrnutý



„The rewriter component of JiST is a dynamic class loader. It intercepts all class load requests and subsequently verifies and modifies the requested classes. These modified, rewritten classes now incorporate the embedded simulation time operations, but they otherwise completely preserve the existing program logic. The program transformations occur once, at load time, and do not incur rewriting overhead during execution. The rewriter also does not require source-code access, since this is a byte code to byte code transformation.“

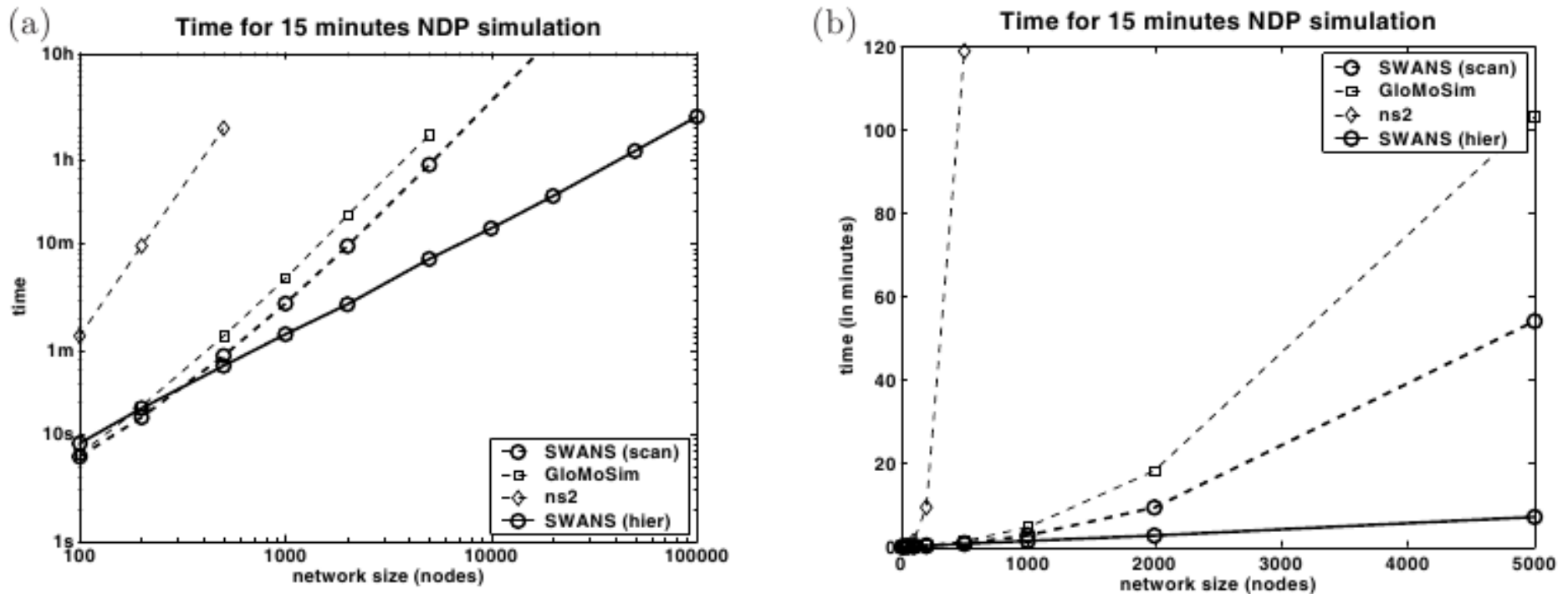


Figure 9. SWANS significantly outperforms both ns2 and GloMoSim in simulations of the node discovery protocol: (a) log-log scale; (b) linear scale.