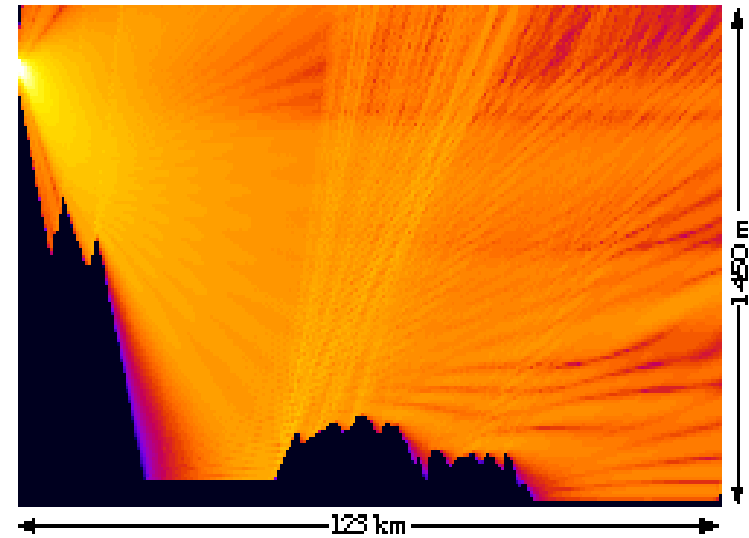
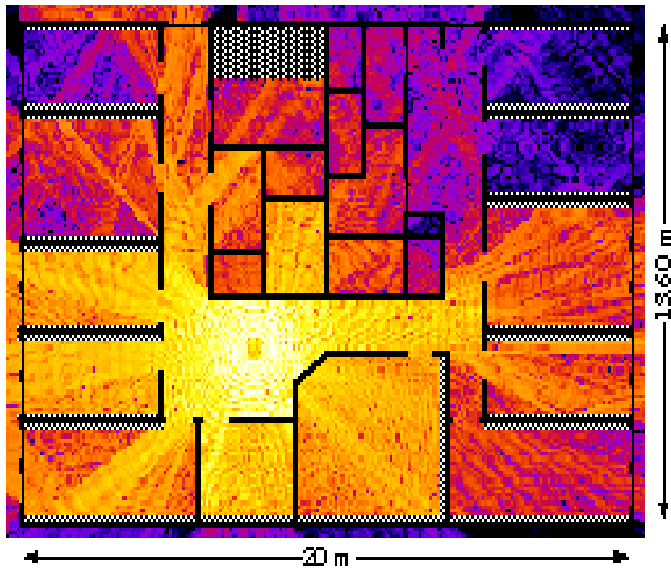


Počítačové siete 2

Šírenie signálu, modelovanie pohybu

Martin Drozda

Šírenie signálu

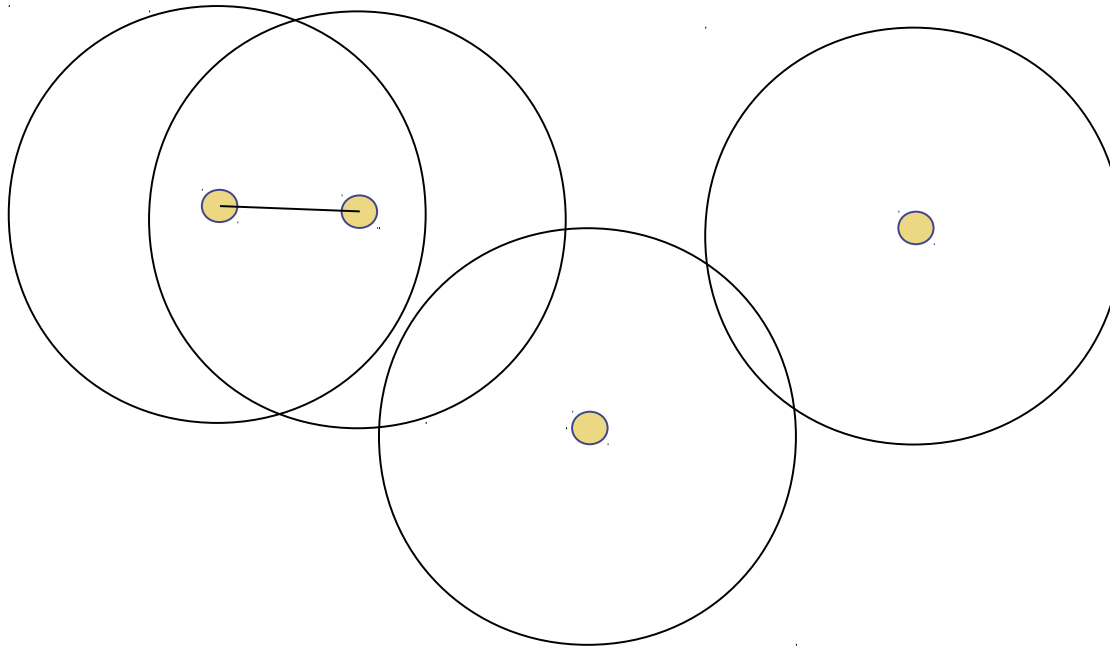


Vypočítat silu signálu pre dané miesto je výpočtovo nemožné.

Zdroj obrázkov: J. Schiller, Mobilkommunikation, SS05, FU BERLIN.

Šírenie signálu

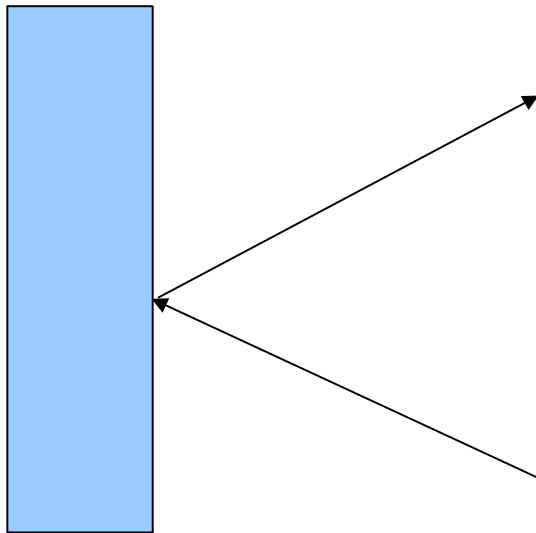
Doteraz sme modelovali šírenie signálu ako kružnicu okolo uzla



... a hranou sme modelovali možnosť posielat' správy medzi dvoma uzlami.

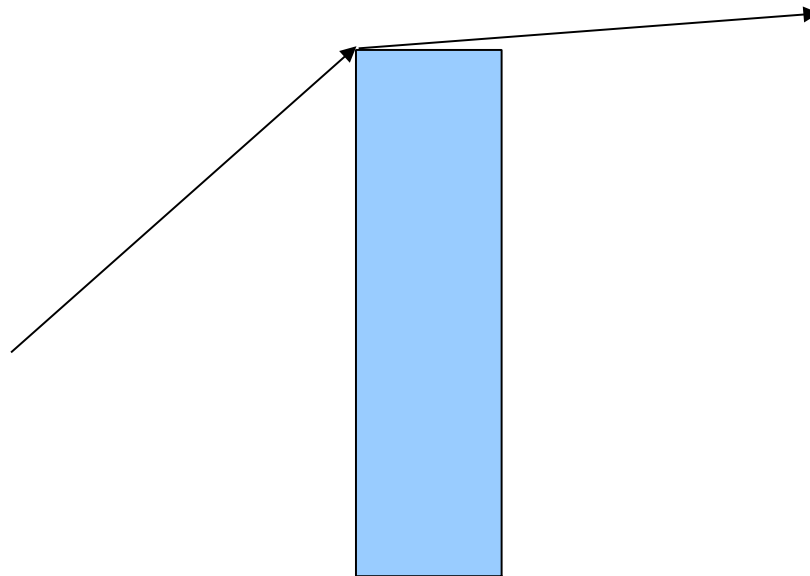
Odraz

Odraz nastane, ak objekt má veľkosť rádovo väčšiu ako vlnová dĺžka.



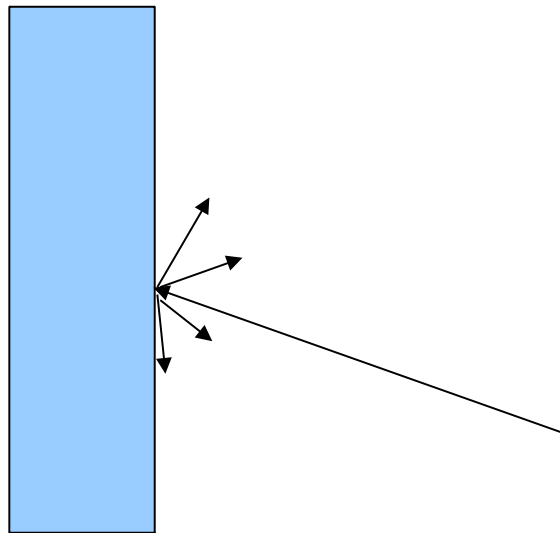
Ohyb (difrakcia)

Ohyb nastane, ak šíreniu signálu prekáža objekt s ostrými hranami. Rádiové vlny zmenia svoj smer, ohnú sa okolo objektu.



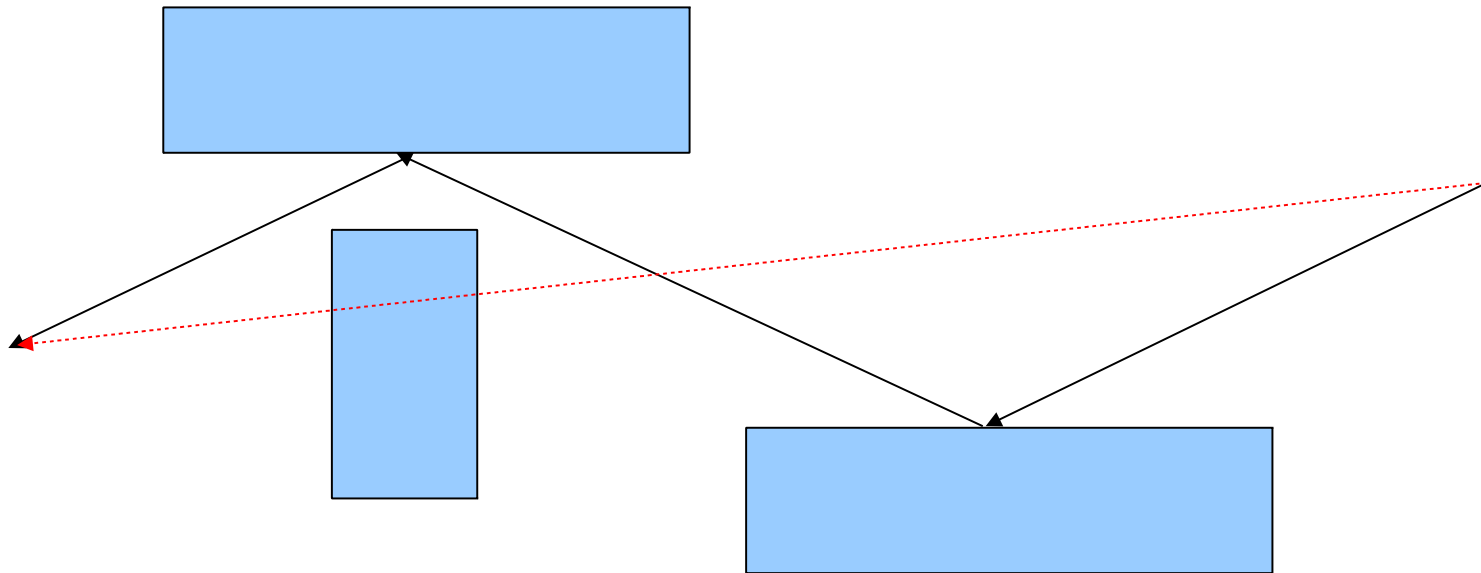
Rozptyl

Rozptyl nastane, ak médium cez, ktoré prechádza rádiová vlna obsahuje objekty porovnateľné alebo menšie ako vlnová dĺžka a počet týchto objektov je veľký. Príklady takýchto objektov sú cestné značenie, stromy s listami, ľudia atď.



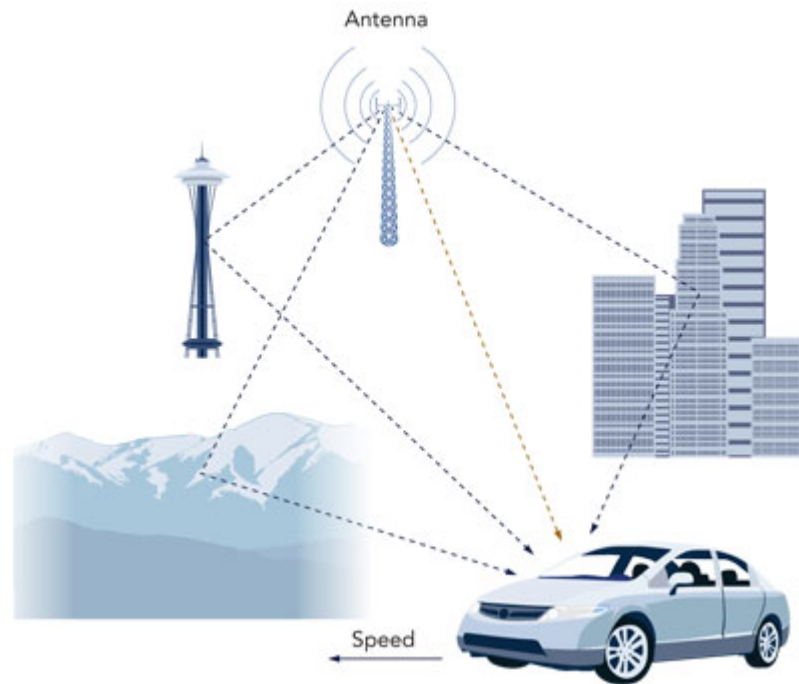
Viac-cestné šírenie (multipath)

V reálnych podmienkach nastáva viac-cestné šírenie signálu, kde sa kombinujú javy ako odraz, ohyb, rozptyl, absorpcia. Tieto javy umožňujú šírenie signálu ináč ako len “priamočiaro” (line of sight).



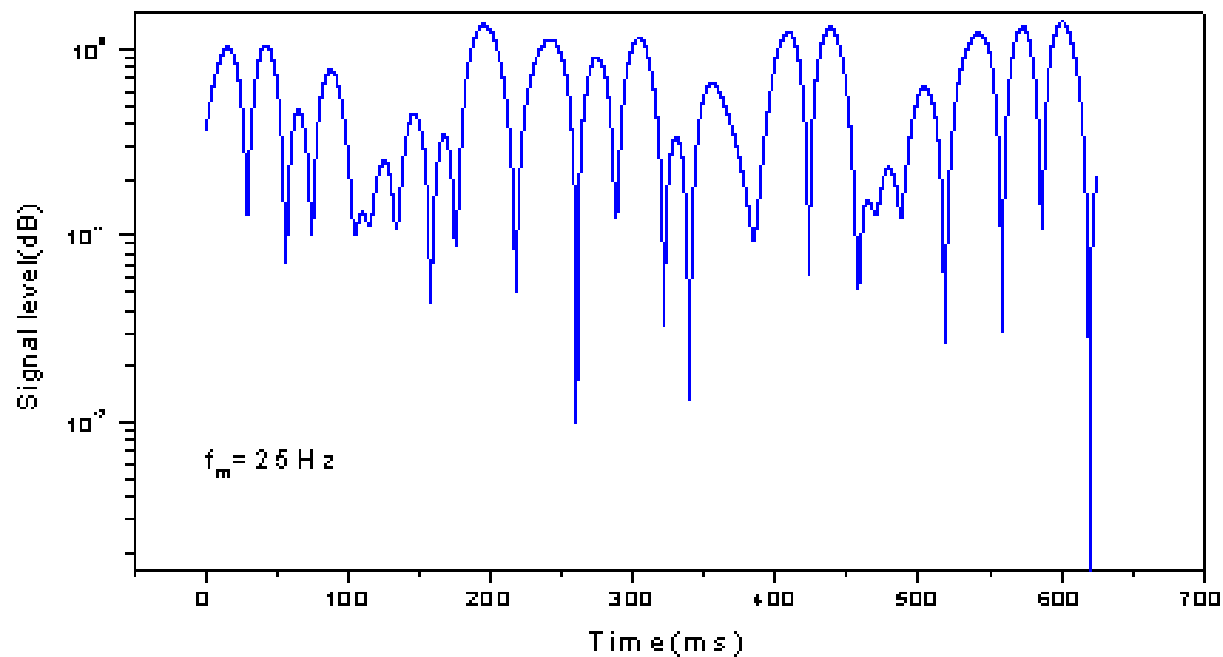
Útlm (fading)

Útlm je dôsledkom odrazu, ohybu, rozptylu a viac-cestného šírenia signálu.



Zdroj obrázku: http://www.tmworld.com/article/509529-RF_fading_simulation.php

Útlm (fading)



Zdroj: M. Djebbouri et al., Estimation of mobile velocity in Rayleigh fading channel. Electronics Letters, 2004.

Rayleighovo rozdelenie

Rayleighovo rozdelenie: štatistický model pre útlm. Rayleighov model útlmu predpokladá, že sila signálu sa mení (utlmuje) podľa Rayleighovho rozdelenia.

Experimentálne merania na Manhattane ukazujú, že útlm v mestkom prostredí je blízky Rayleighovmu modelu.

Ak existuje dominantné šírenie signálu bez prekážok potom je Riceanovský model presnejší.

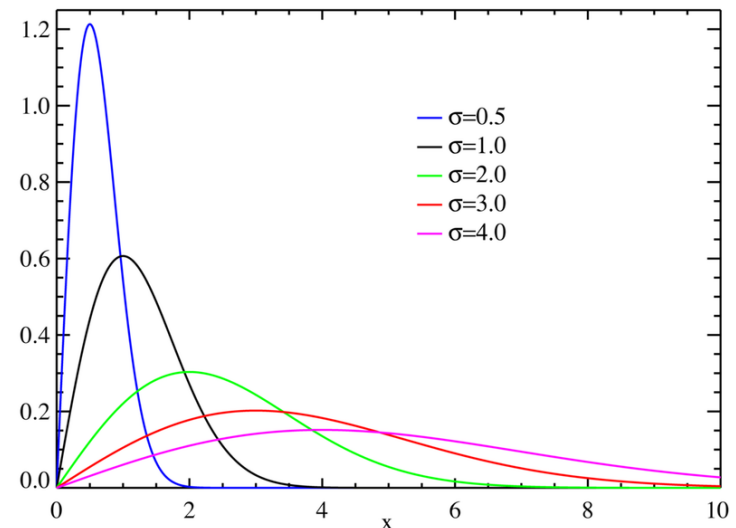
Rayleigh distribution:

$$p(x) = \frac{x}{\sigma^2} e^{-\frac{x^2}{2\sigma^2}}; 0 \leq x \leq \infty$$

Mód: σ

Stredná hodnota: $\sigma \sqrt{\pi/2}$

Rozptyl: $\sigma^2 (4-\pi)/2$



Jednotky:

W	dB	relatívne k 1W	
mW	dBm	relatívne k 1mW	(“dee bee milliwatt”)

Prevod:

$$P(\text{dBm}) = 10 \log (P(\text{mW})/1\text{mW})$$

$$P(\text{mW}) = 50\text{W}$$

$$P(\text{dBm}) = 10 \log (50 \cdot 10^3) = 47.0 \text{ dBm}$$

$$P(\text{mW}) = 5\text{mW}$$

$$P(\text{dBm}) = 10 \log (5) = 6.9 \text{ dBm}$$

$$P(\text{mW}) = 1\text{mW}$$

$$P(\text{dBm}) = 10 \log (1) = \underline{0 \text{ dBm}}$$

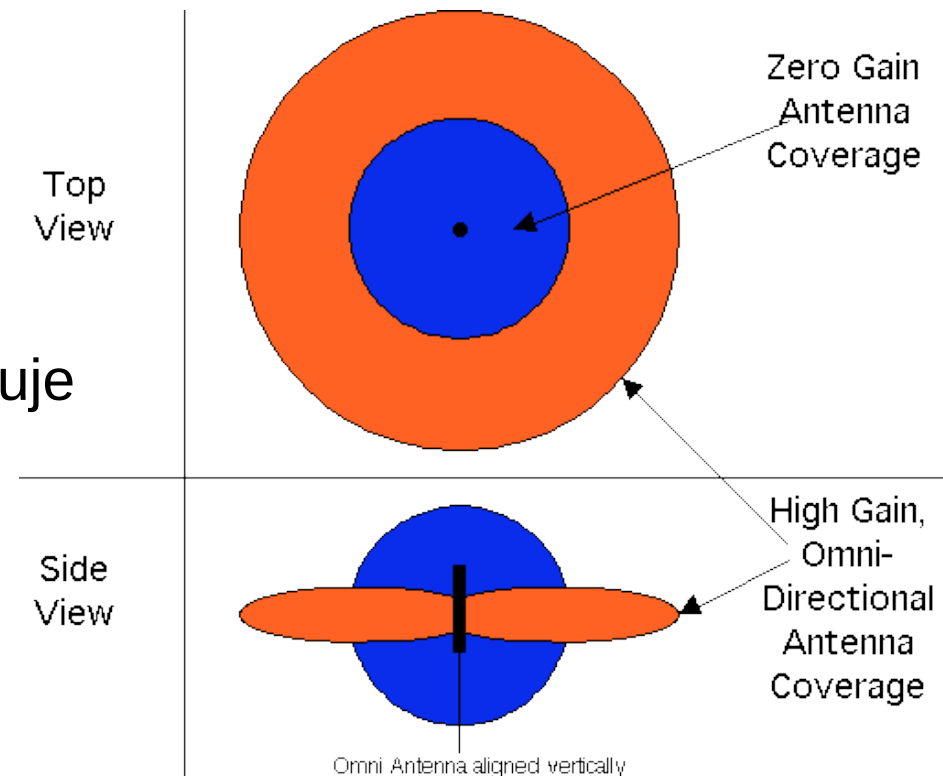
Izotropná anténa: teoretická ideálna anténa, ktorá vyžaruje rovnako všetkými smermi.

Nesmerová (omnidirectional): vyžaruje „rovnako“ všetkými smermi.

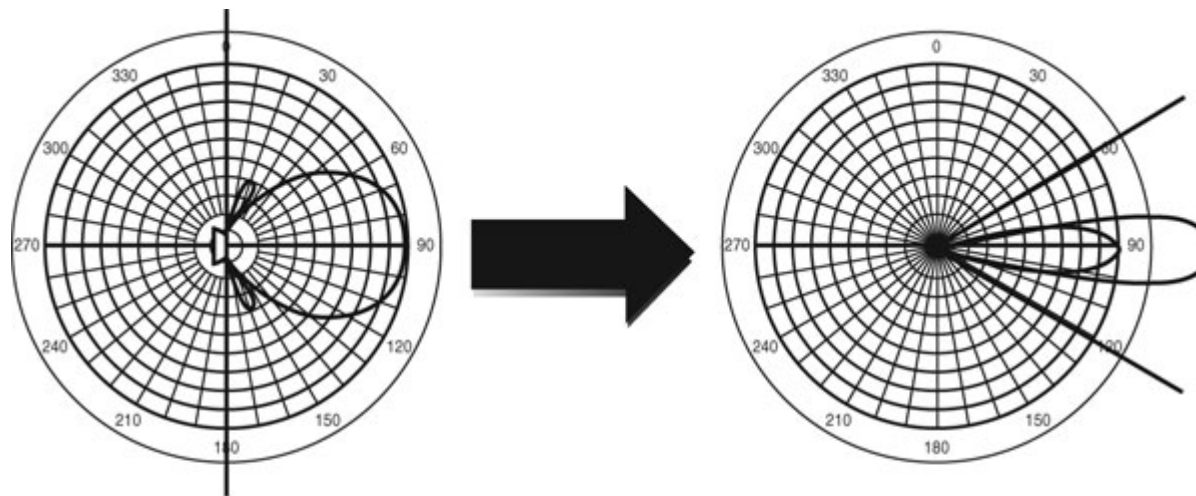
Smerová: vyžaruje jedným smerom.

Zisk antény:

- Relatívne k izotropnej anténe v Db.
- Zisk antény je definovaný ako pomer intenzity vyžarovania v danom smere a vyžarovania izotropnej antény. Vyjadruje tiež „smerovosť“ antény.
- Zisk izotropnej antény 0dB = jednotkový zisk.



Zisk antény



Use of directors makes angle smaller and gain higher.

By using a reflector all the energy appears in only one hemisphere and that results in a doubling of radiated energy in this direction or 3dB gain.

Free-space model šírenia

Šírenie signálu bez akýchkoľvek prekážok

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}$$

$P_r(d)$ = sila signálu pri prijímači vo vzdialenosti d od vysielača

P_t = sila signálu pri vysielači

G_t, G_r = zisk antény (vysielač, prijímač)

λ = vlnová dĺžka

d = vzdialenosť vysielač-prijímač

L = hardvérové straty $L \geq 1$

$\lambda = c/f$; c = rýchlosť svetla; f = frekvencia

Free-space model šírenia

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}$$

$$P_r(d) = 4.94 \times 10^{-7} \text{ W} = -33 \text{ dBm}$$

$P_t = 50\text{W}$ (bežné wifi -20dBm, teda menej ako 1mW)

$G_t, G_r = 0\text{dB}$ (jednotkový zisk)

$\lambda = 0.125\text{m}$ (2.4GHz)

$d = 100\text{m}$

$L = 1$

Free-space model šírenia

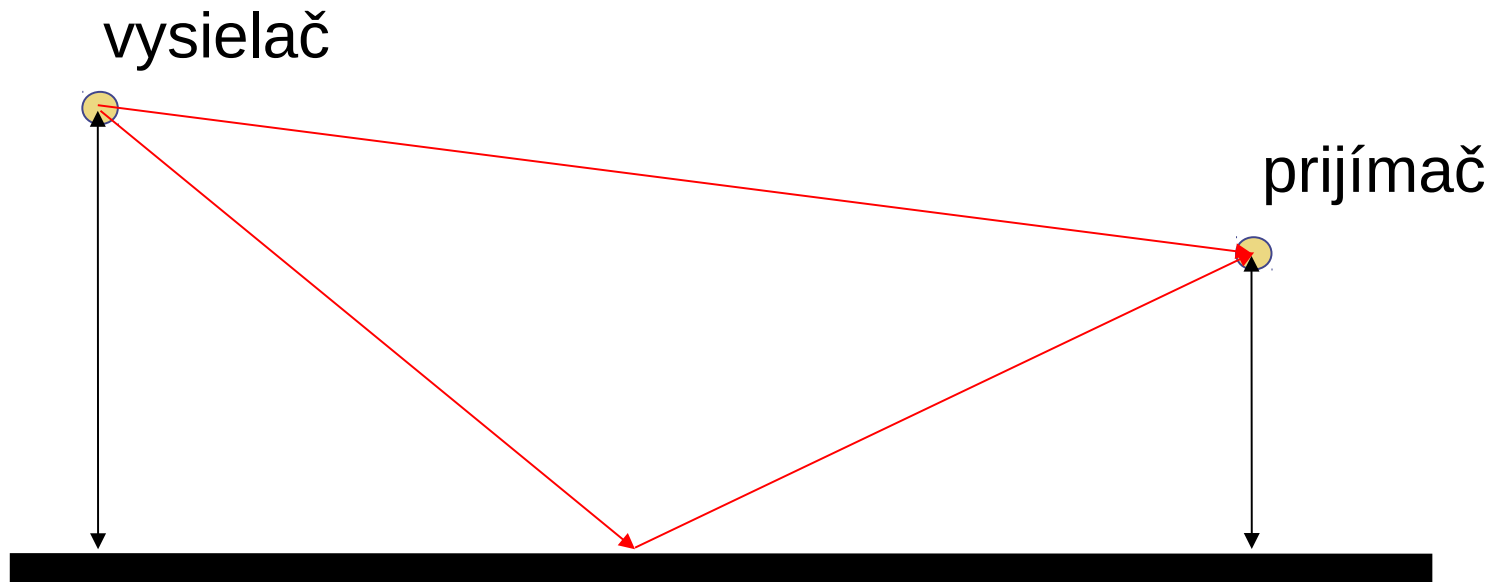
$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{(4\pi)^2 d^2 L}$$

Receiver sensitivity: -91dBm
Receiver threshold: -81dBm

Distance $d = ???$

Two-ray model šírenia

Vrátane odrazu od zeme



Two-ray model šírenia

$$P_r(d) = \frac{P_t G_t G_r h_t^2 h_r^2}{d^4}$$

h_t = výška vysielачa
 h_r = výška prijímača

Predpoklad: $d \gg h_t + h_r$ a perfektný odraz od zeme (bez rozptylu a absorpcie)

Nastavenia simulátora

```
/** Default radio frequency (units: Hz). */  
public static final double FREQUENCY_DEFAULT = 2.4e9; // 2.4 Ghz  
  
/** Default radio bandwidth (units: bits/second). */  
public static final int BANDWIDTH_DEFAULT = (int)1e6; // 1Mb/s  
  
/** Default transmission strength (units: dBm). */  
public static final double TRANSMIT_DEFAULT = 15.0;  
  
/** Default antenna gain (units: dB). */  
public static final double GAIN_DEFAULT = 0.0;  
  
/** Default radio reception sensitivity (units: dBm). */  
public static final double SENSITIVITY_DEFAULT = -91.0;  
  
/** Default radio reception threshold (units: dBm). */  
public static final double THRESHOLD_DEFAULT = -81.0;
```

Vzdialenostný útlm je zníženie sily signálu počas jeho šírenia

$$PL(d) = PL(d_0) + 10n \log(d / d_0)$$

Odhad vzdialenostného útlmu:

d_0 = referenčná vzdialenosť

n = exponent (path-loss exponent):

Free space (bez prekážok)	2
Mesto	2.7-3.5
V budove bez prekážok	1.6-1.8
V budove s prekážkami	4-6
Vo výrobnjej hale	2-3

Tunel môže pôsobiť ako vlnovod (waveguide) a mať exponent menší ako 2.

ns3::MatrixPropagationLossModel

```
Ptr<MatrixPropagationLossModel> lossModel =  
CreateObject<MatrixPropagationLossModel> ();
```

```
lossModel->SetDefaultLoss (200); // set default loss to 200 dB
```

```
lossModel->SetLoss (nodes.Get (0)->GetObject<MobilityModel> (),  
nodes.Get (1)->GetObject<MobilityModel> (), 50); // set symmetric  
loss 0 <-> 1 to 50 dB
```

```
lossModel->SetLoss (nodes.Get (2)->GetObject<MobilityModel> (),  
nodes.Get (1)->GetObject<MobilityModel> (), 50); // set symmetric  
loss 2 <-> 1 to 50 dB
```

MatrixPropagationLossModel: nie je závislý od vzdialenosti

Ďalšie:

TwoRayGroundPropagationLossModel

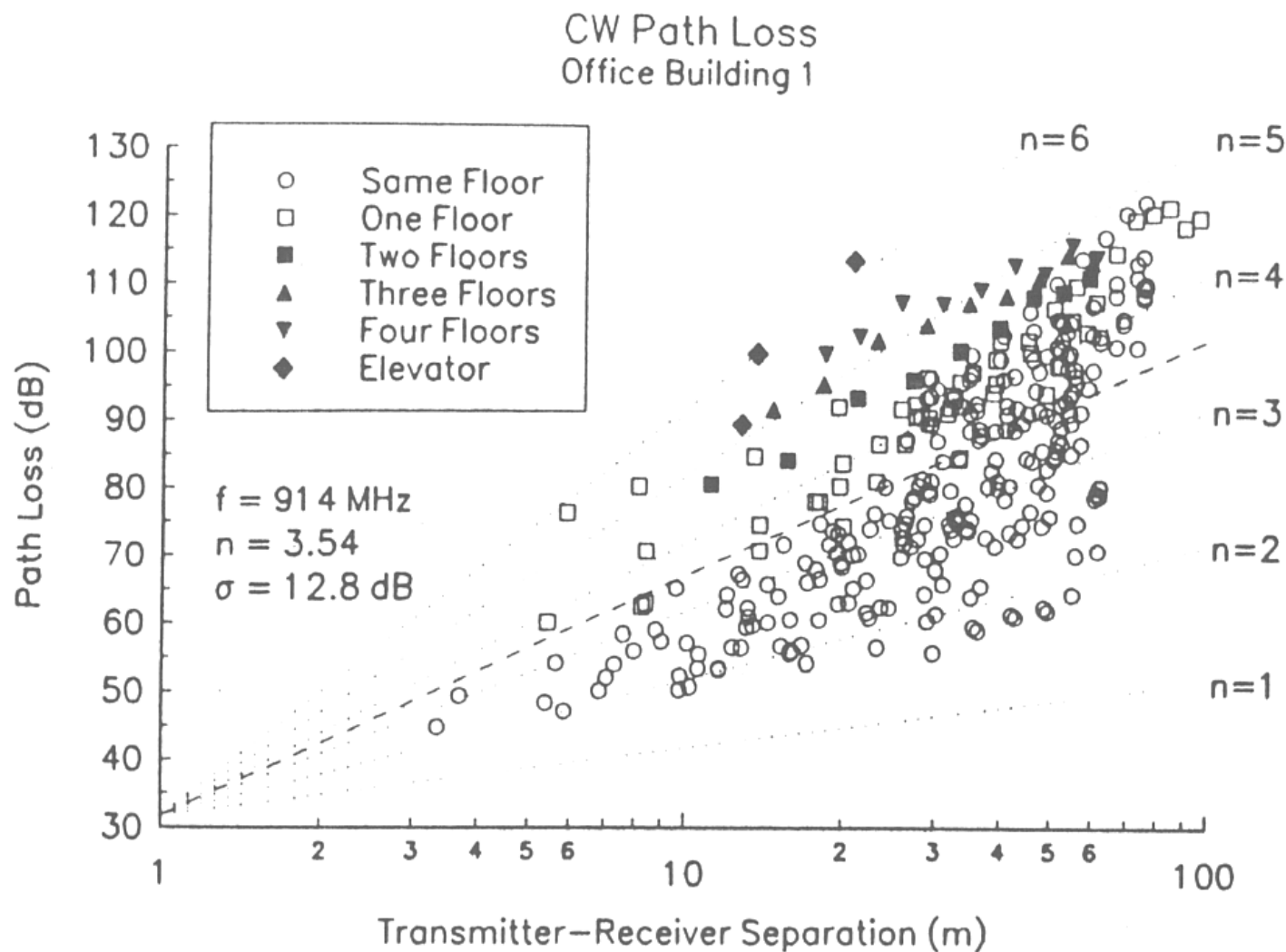
FriisPropagationLossModel: Free space model

```
Ptr<TwoRayGroundPropagationLossModel> lossModel =  
CreateObject<TwoRayGroundPropagationLossModel> ();  
  
wifiChannel->SetPropagationLossModel(lossModel);  
wifiChannel->SetPropagationDelayModel (CreateObject  
<ConstantSpeedPropagationDelayModel> ());
```

A ešte nastaviť potrebné parametre:

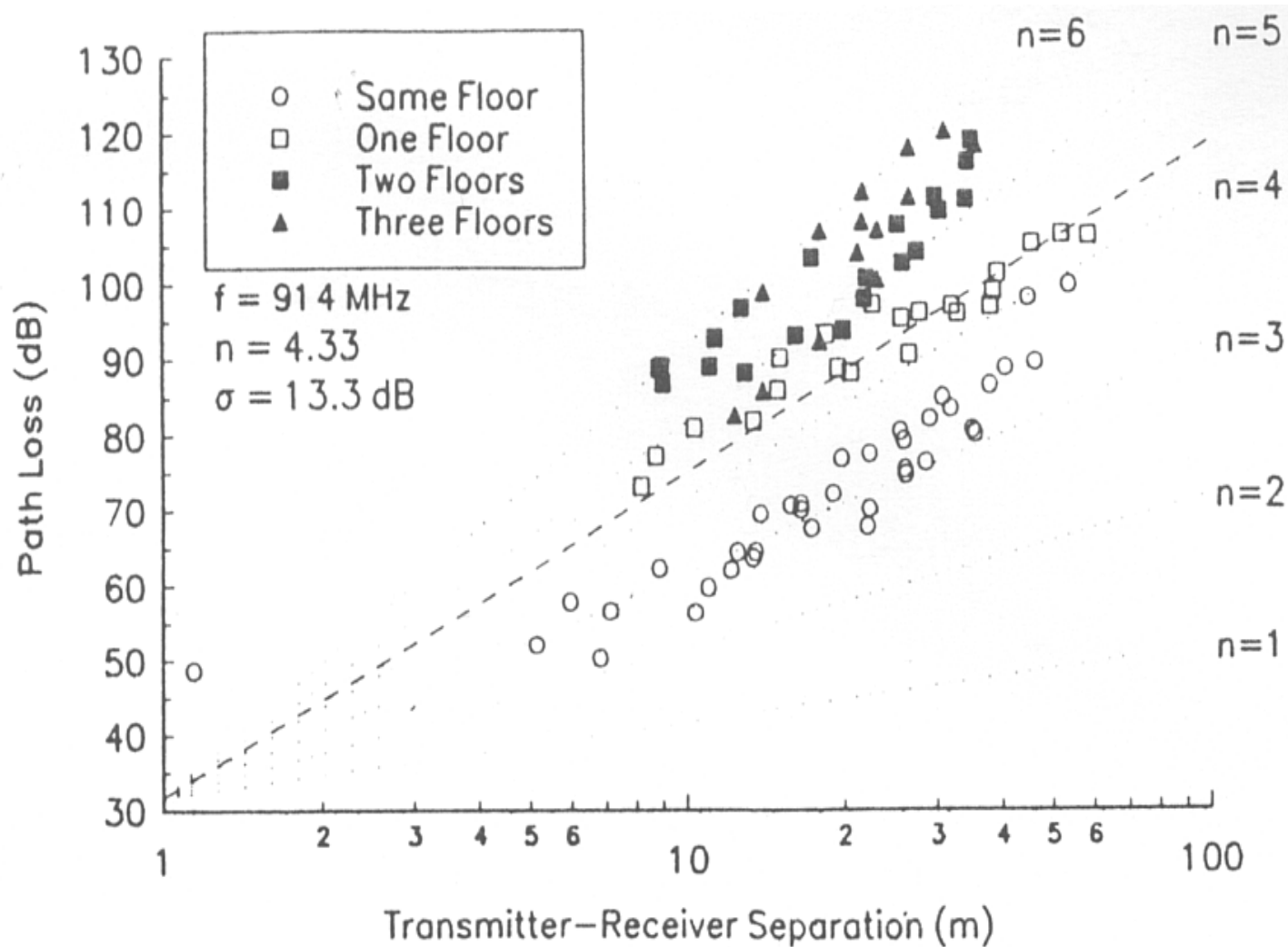
```
double    GetFrequency (void) const  
double    GetMinDistance (void) const  
double    GetSystemLoss (void) const  
  
void      SetFrequency (double frequency)  
void      SetHeightAboveZ (double heightAboveZ)  
void      SetMinDistance (double minDistance)  
void      SetSystemLoss (double systemLoss)
```

Vzdialenostný útlm v budove



Zdroj: www.sss-mag.com/indoor.html#tutorial, 1998
 σ = smerodajná odchýlka vzdialenostného útlmu

Vzdialenostný útlm v budove



Zdroj: www.sss-mag.com/indoor.html#tutorial, 1998

Realistické modelovanie pohybu:

- *Individuálne modely: každý uzol je modelovaný zvlášť.*
- *Skupinové modely: modeluje niekoľko uzlov ako jednu entitu.*

Mnoho modelov pohybu je už implementovaných v sieťových simulátoroch.

Cieľ bolo namodelovať výrazne náhodný pohyb.

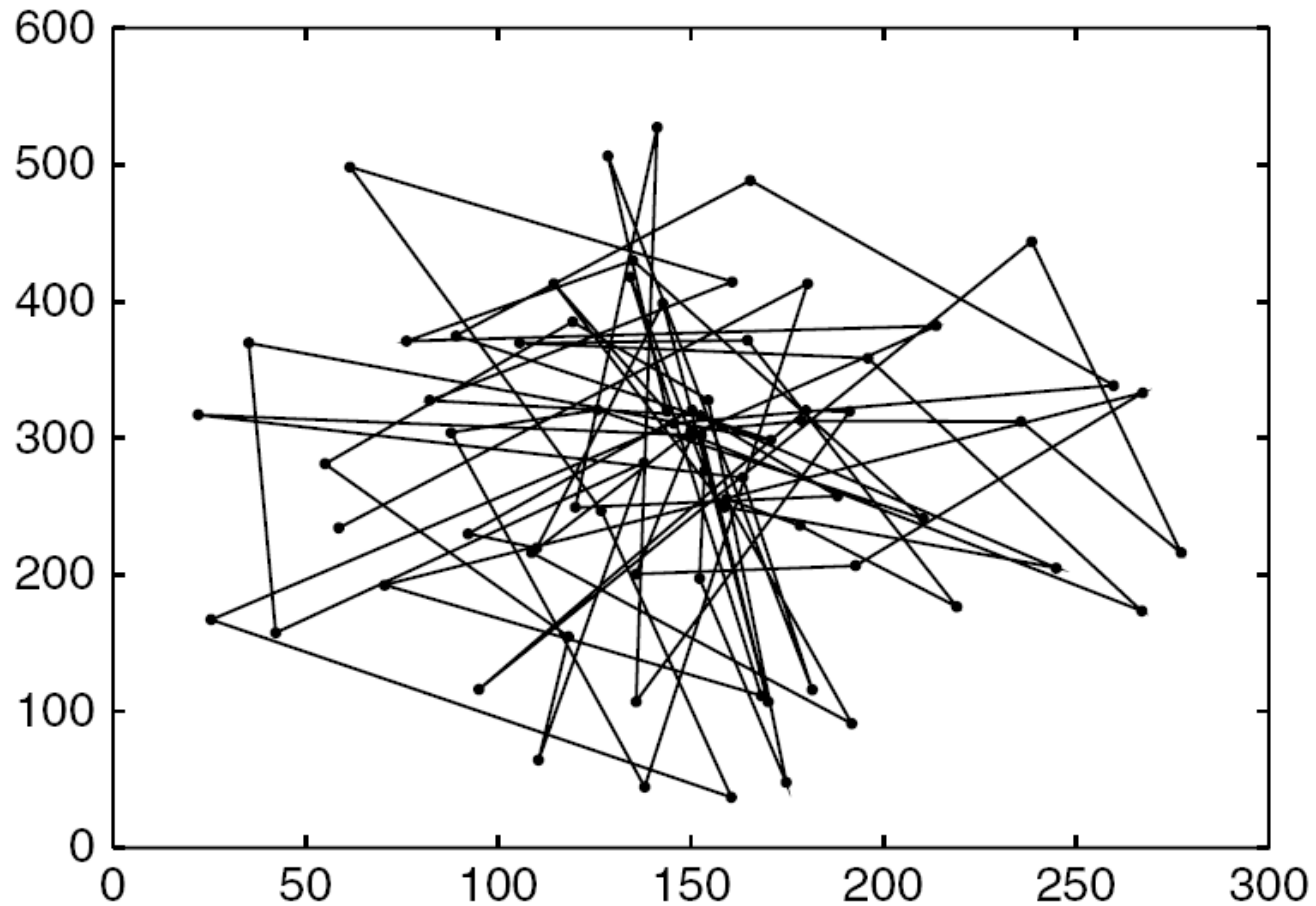
Uzly sa pohybujú z terajšej lokácie do novej tak, že si náhodne zvolia smer a rýchlosť.

$\alpha = -\pi \dots +\pi$
 $\text{speed} = \text{speed}_{\min} \dots \text{speed}_{\max}$

Nová lokácia je zvolená buď po danom čase alebo vzdialenosti.

Model nemá pamäť: nepamätá si predchádzajúci pohyb.

Random walk



Speed= 0 ... 10 m/s; time = 60s

Random waypoint model

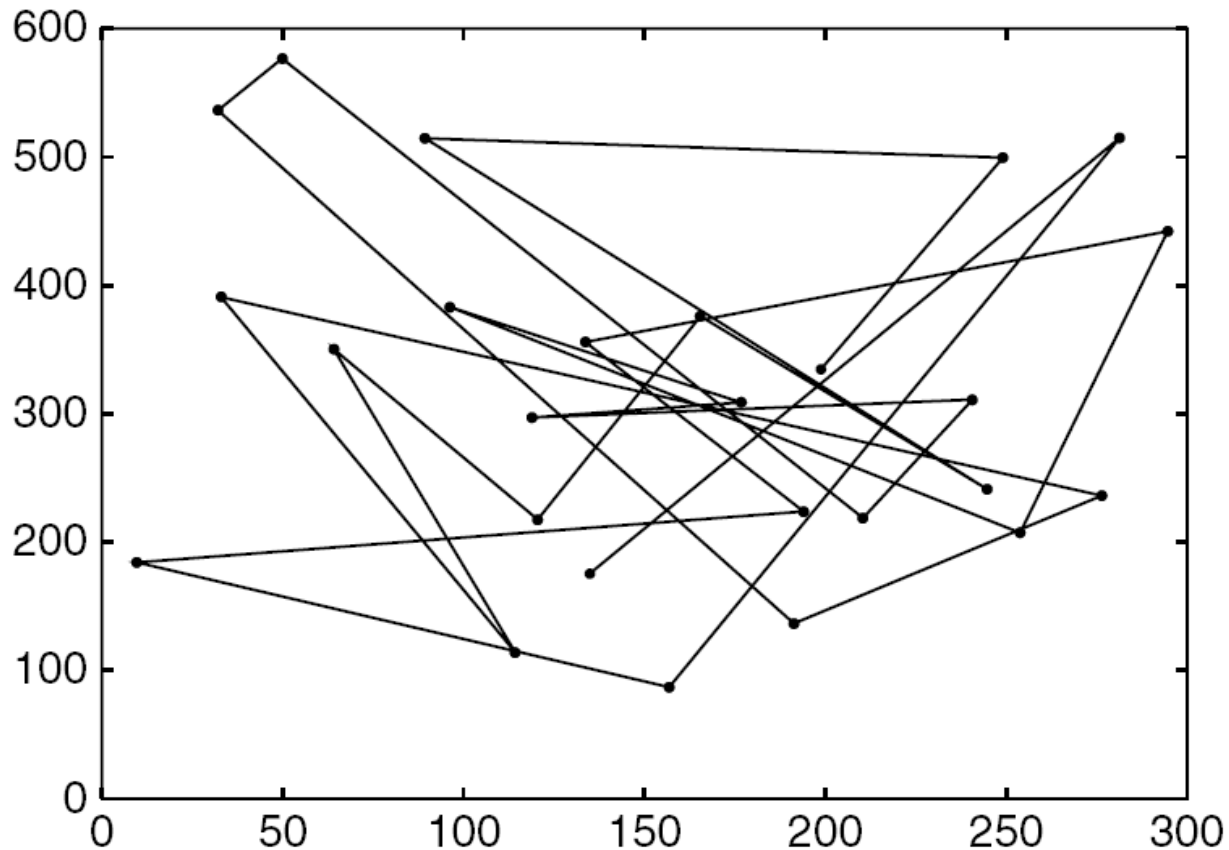
Uzly sa pohybujú z terajšej lokácie do novej tak, že si náhodne vyberú novú lokáciu a rýchlosť.

$new(x,y)$ = vrámci simulovanej plochy
 $speed = speed_{min} \dots speed_{max}$
 $pause = pause_{min} \dots pause_{max}$

Nová lokácia je zvolená po dosiahnutí terajšej lokácie. Na lokácii je možné zotrvať (pause).

Model nemá pamäť: nepamätá si predchádzajúci pohyb.

Random waypoint model



speed = 0 ... 10 m/s

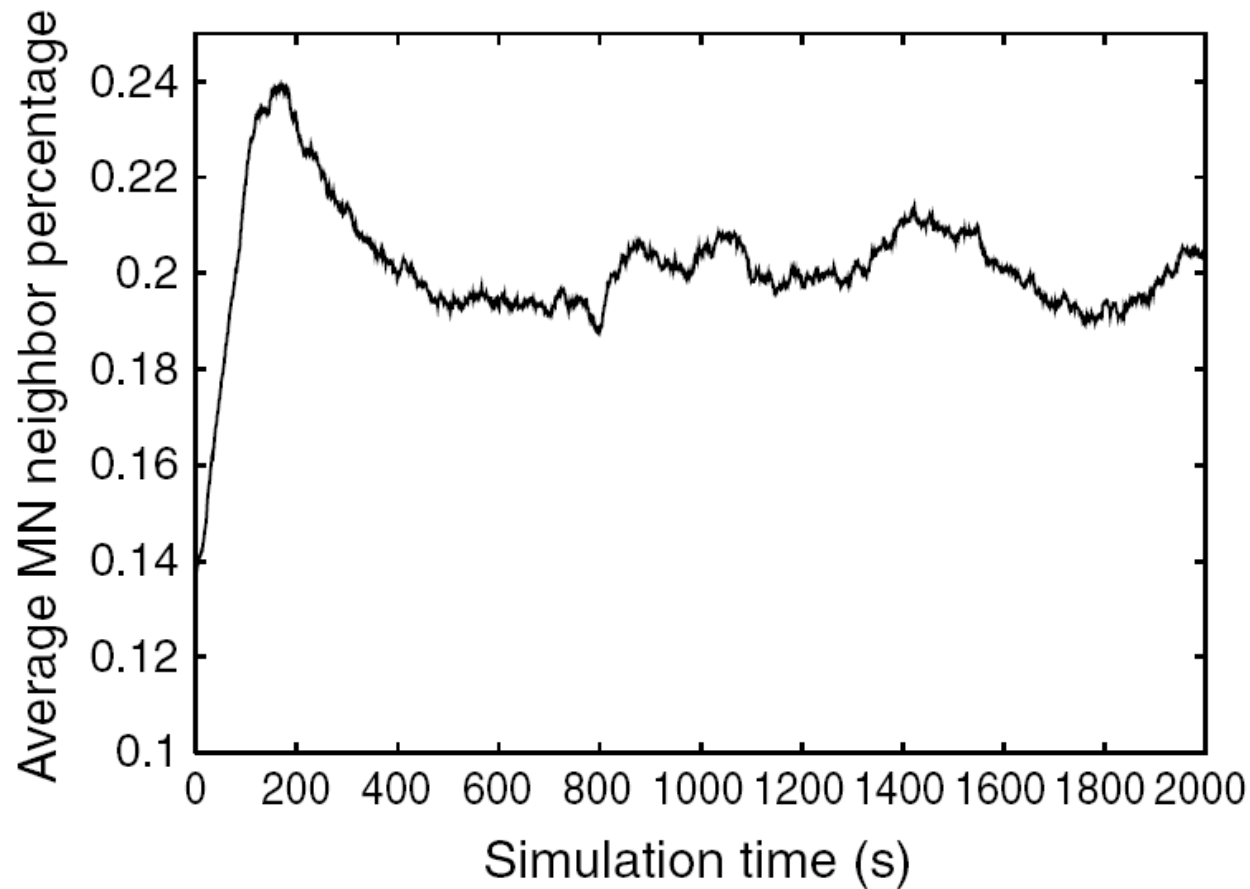
Model má tendenciu zhukovania uzlov v strede simulovanej oblasti

Random waypoint model: vlastnosti

1. **Zhlukovanie uzlov** v strede.

2. **Spomaľovanie uzlov**: existuje veľká pravdepodobnosť, že si uzol zvolí ďalekú lokáciu a zároveň nízku rýchlosť.

Random waypoint model



Random direction model

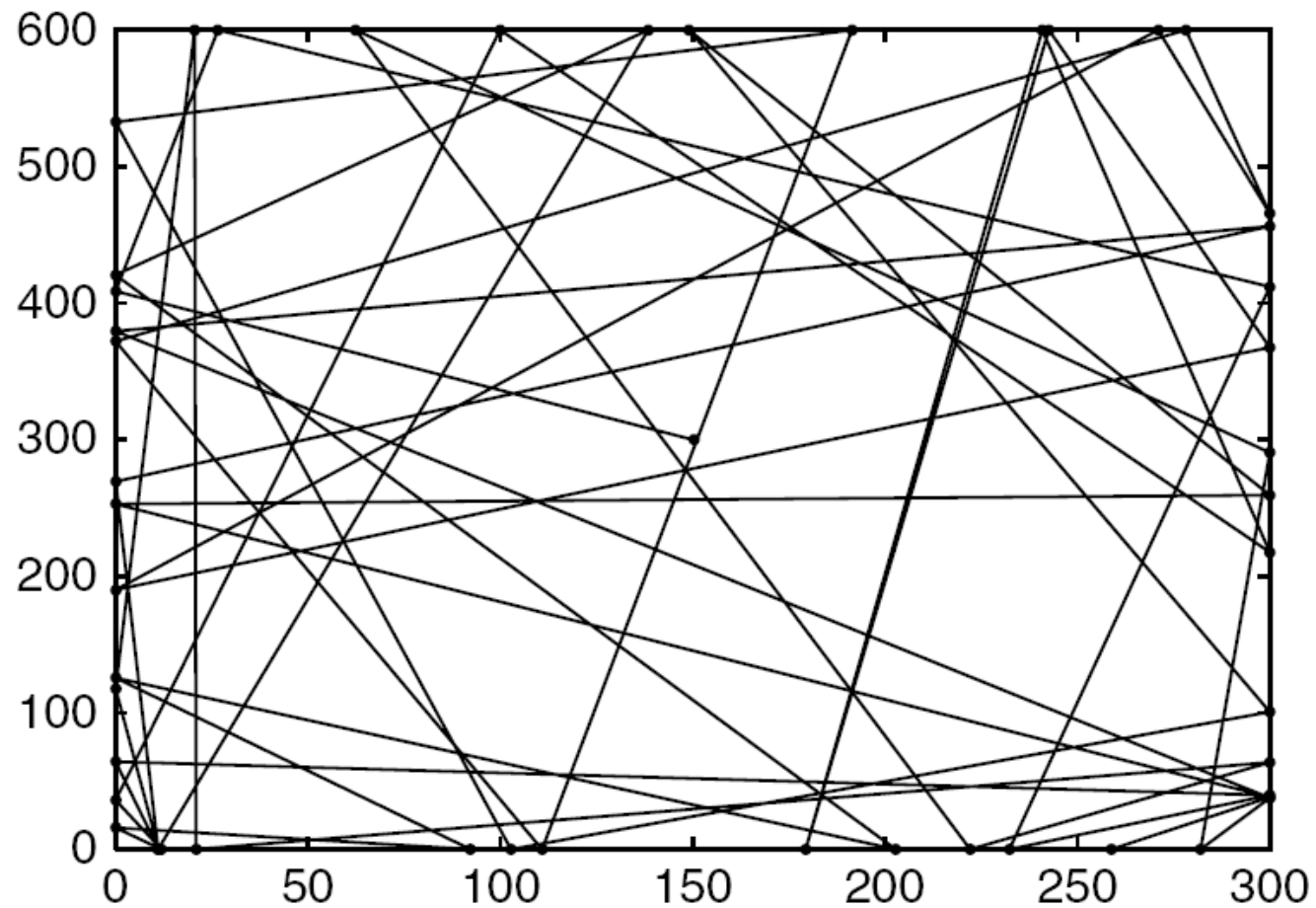
Uzly si zvolia náhodný smer.

$$\alpha = 0 \dots +\pi$$

Model nemá pamäť: nepamätá si predchádzajúci pohyb.

Žiadne zhlukovanie v strede.

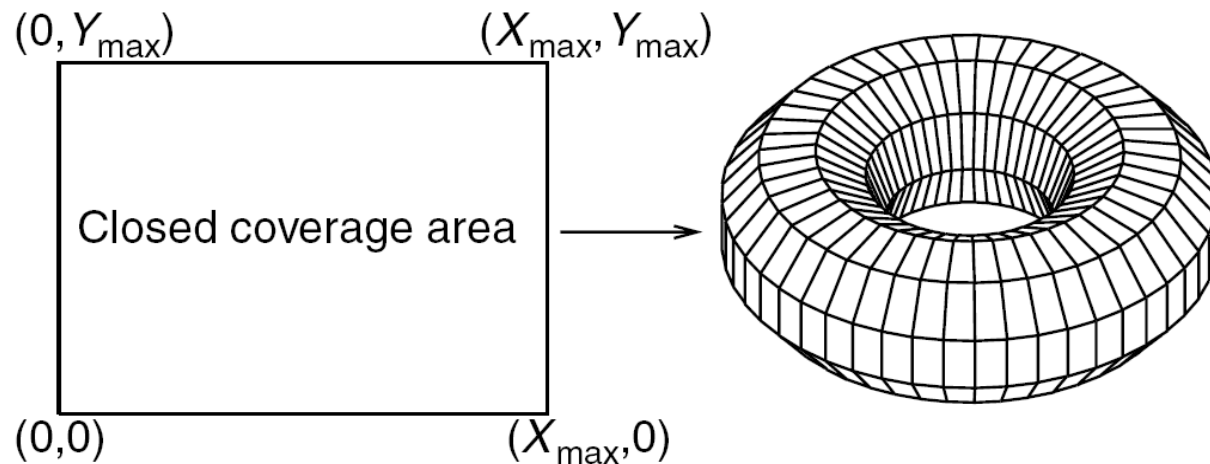
Random direction model



Simulovaná oblasť bez hraníc

Po dosiahnutí hranice sa uzol zjaví na druhej strane.

Ako modelovať šírenie signálu v tomto prípade?



Gauss-Markov model pohybu

$$s_n = \alpha s_{n-1} + (1 - \alpha)\bar{s} + \sqrt{(1 - \alpha^2)s_{x_{n-1}}}$$

$$d_n = \alpha d_{n-1} + (1 - \alpha)\bar{d} + \sqrt{(1 - \alpha^2)d_{x_{n-1}}}$$

s_n = rýchlosť pre krok n

d_n = smer pre krok n

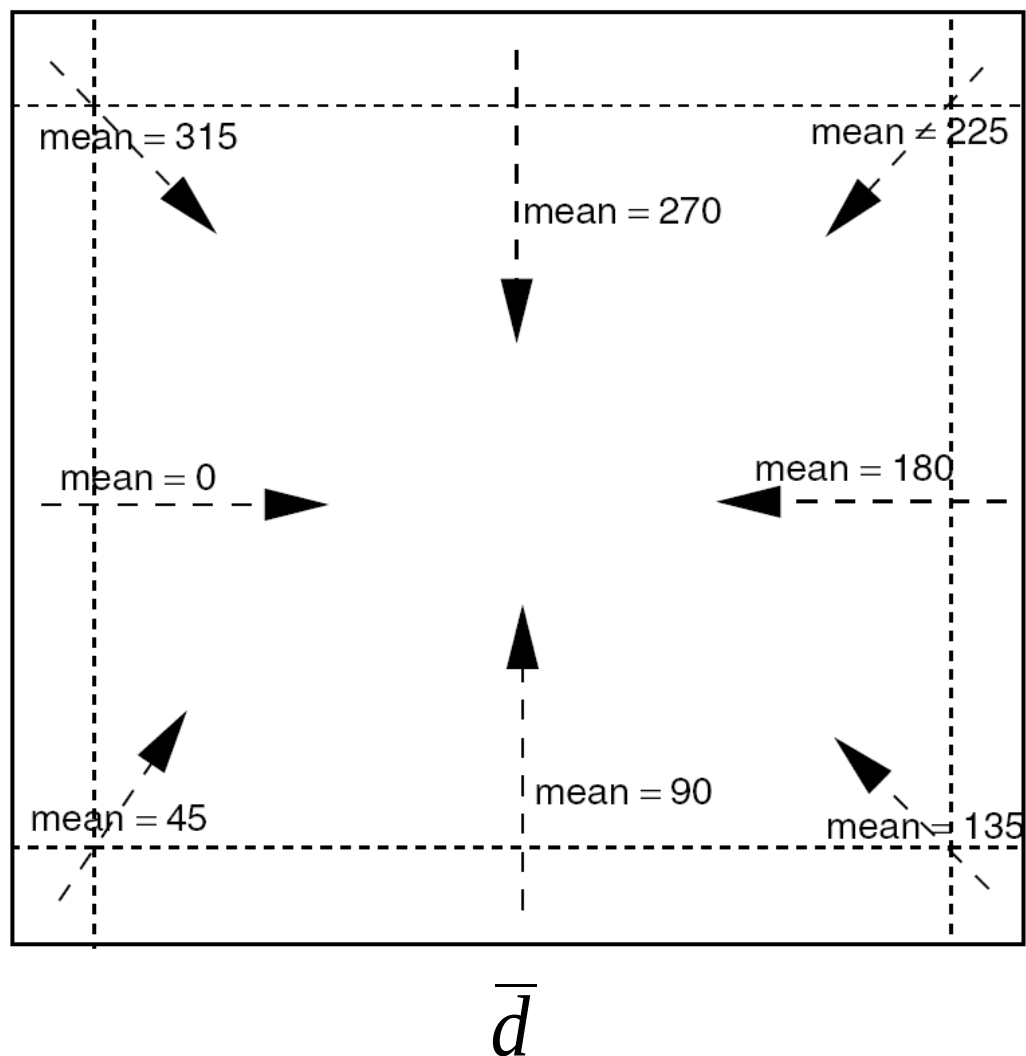
$\bar{d}$ = odraz od hranice (v radiánoch)

$s_{x_{n-1}}, d_{x_{n-1}}$ = náhodné premenné s normálnym rozdelením

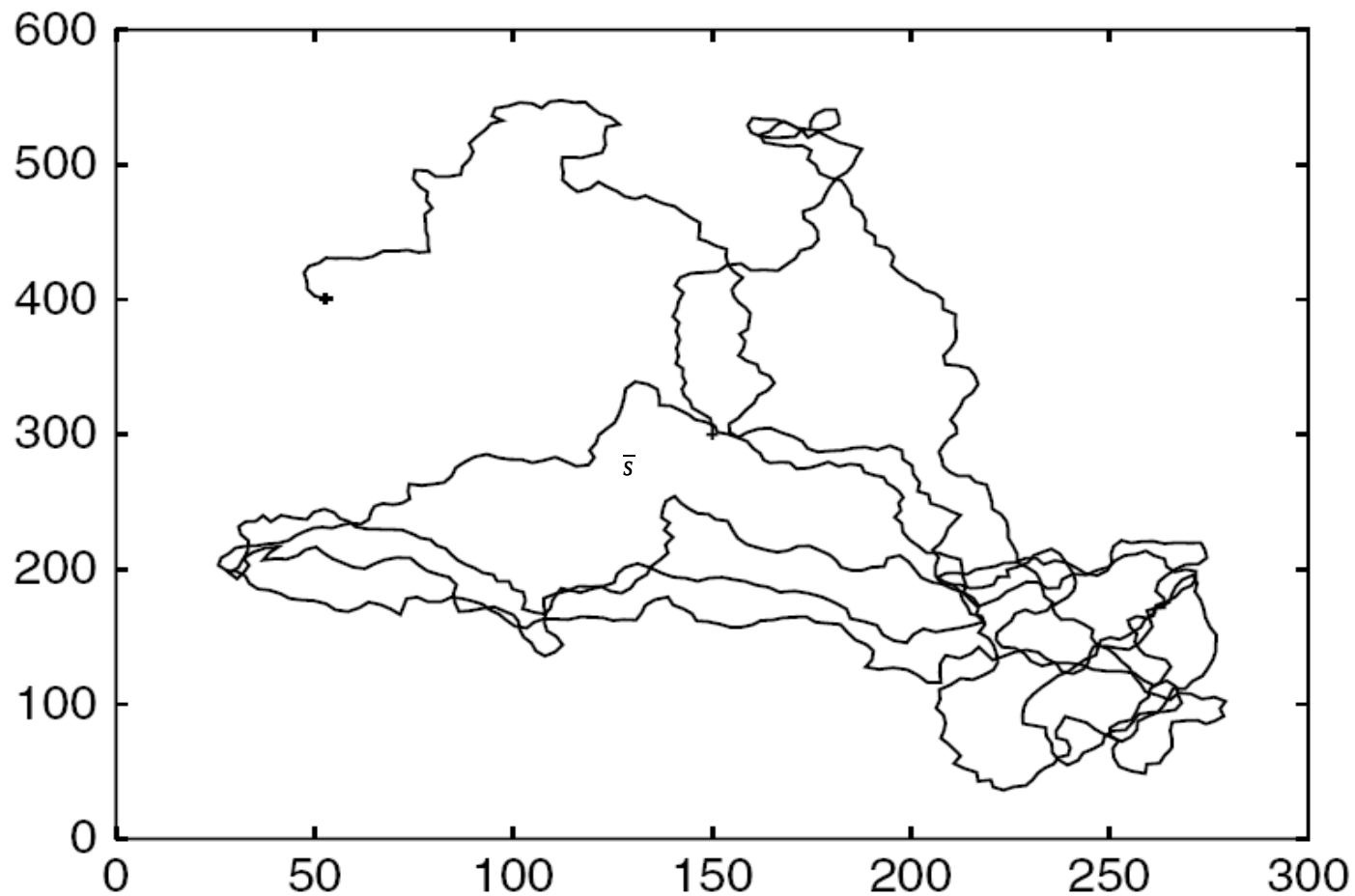
$\alpha=0$, Brownovský pohyb

$\alpha=1$, priamočiary pohyb

Gauss-Markov model pohybu

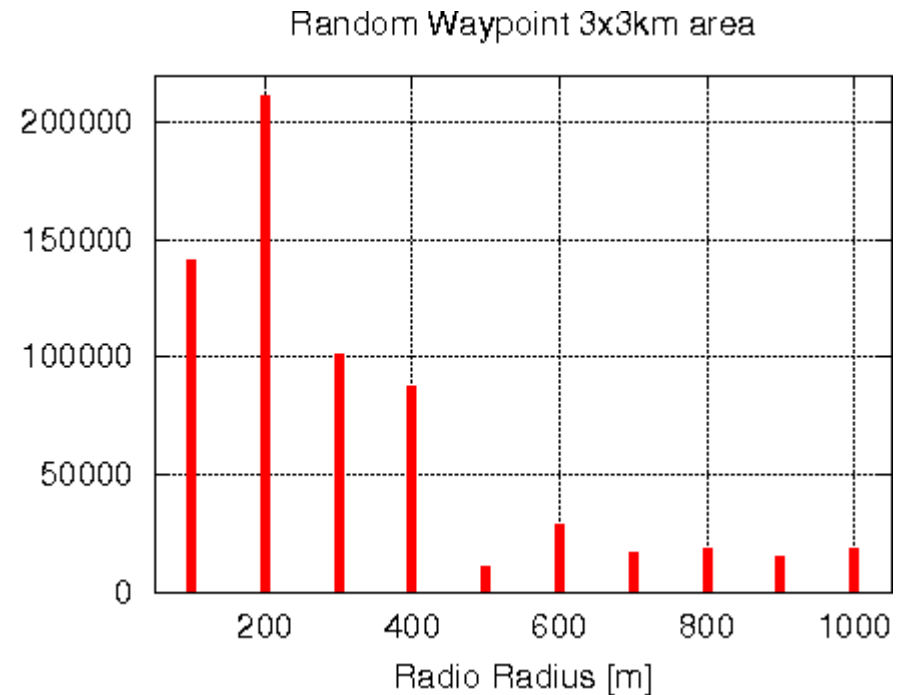
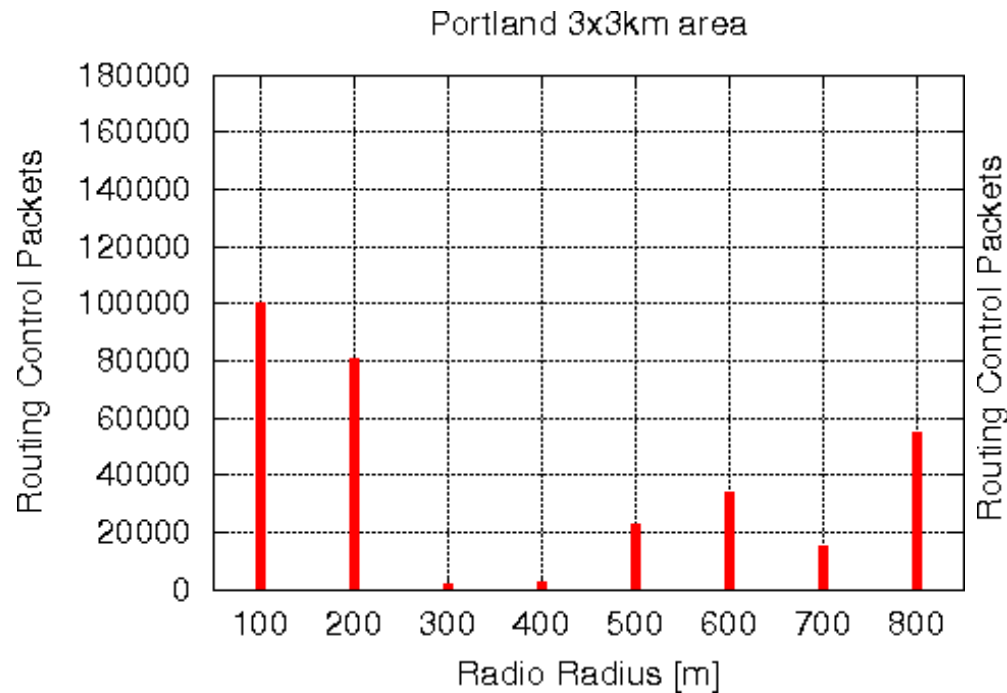


Gauss-Markov model pohybu



$$\bar{s} = 10 \text{ m / s}$$

Realistické modely pohybu



Rozdielne d2-párovanie, rozdielne rozdelenie najkratších ciest, priemeru atď. Dôsledok: iná výkonnosť protokolov.

Exponential correlated random mobility model

$$b(t + 1) = b(t)e^{-\frac{1}{\tau}} + \left(\sigma \sqrt{1 - \left(e^{-\frac{1}{\tau}} \right)^2} \right) r$$

τ = parameter zmeny (konštanta)

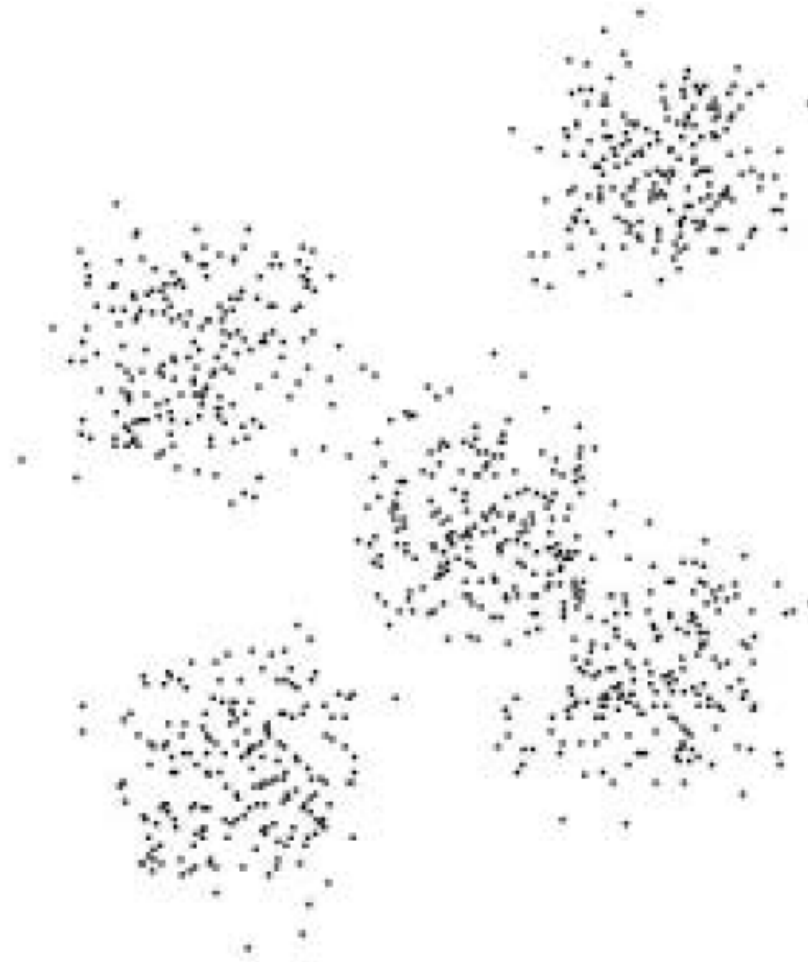
r = náhodná premenná s normálnym rozdeleným a štandardnou odchýlkou σ

$b(t)$ môže byť uzol alebo skupina uzlov

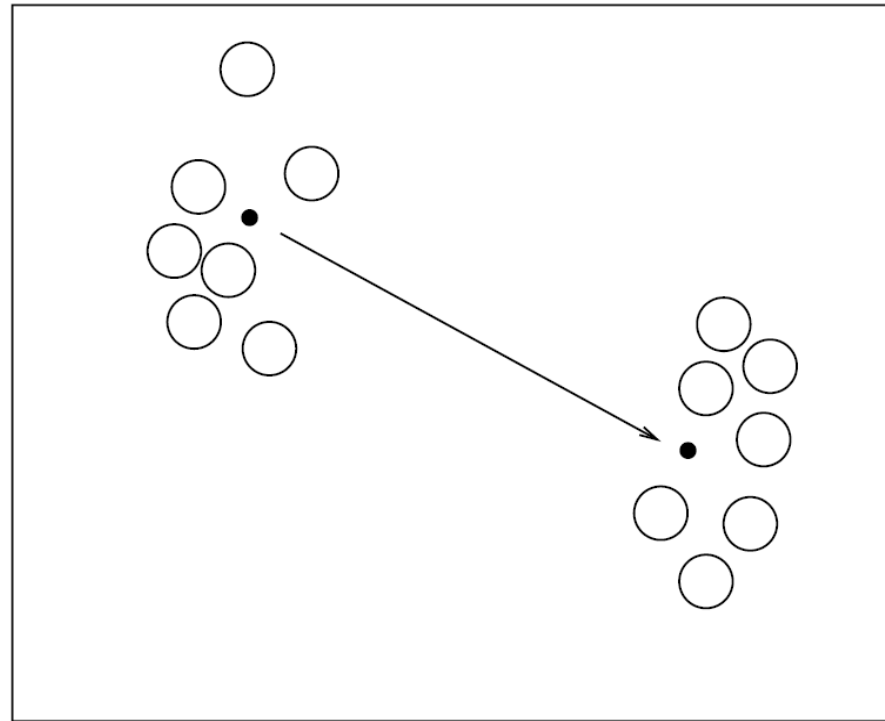
Exponential correlated random mobility model

1 středový bod pre skupiny

5 středových bodov pre každú skupinu



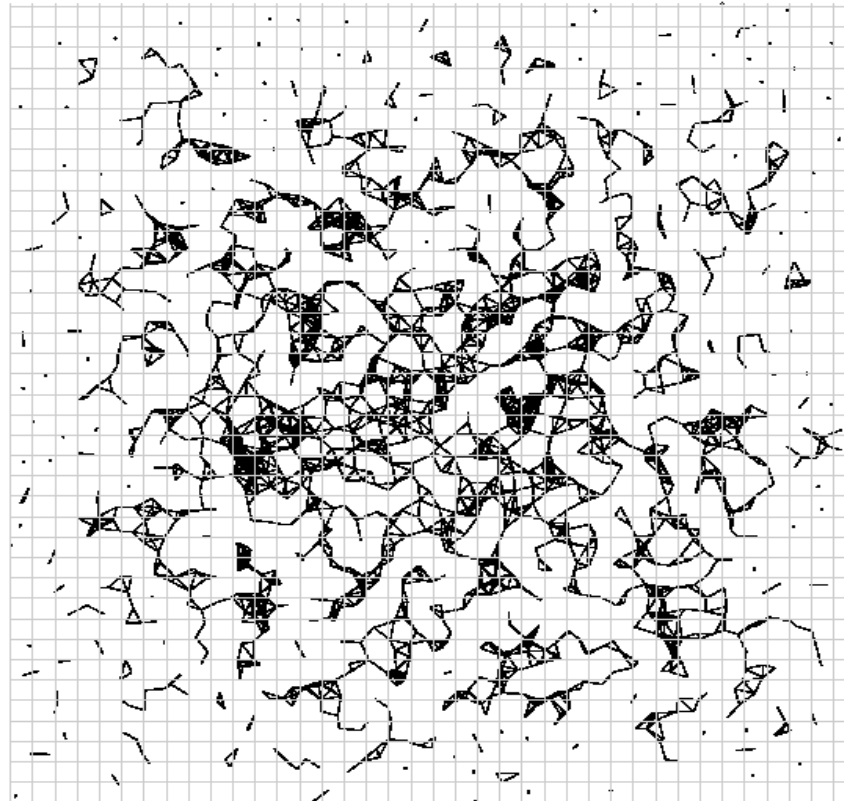
Nomadic community mobility model



Uzly sa pohybujú spolu v skupinách.

Každý uzol používa Random walk model na pohyb okolo centra skupiny.

Statické siete



Random waypoint zastavený
po čase t

ConstantPositionMobilityModel

```
NodeContainer nodes;  
nodes.Create(2);  
  
for (size_t i = 0; i < 2; ++i){  
    nodes.Get(i)->  
        AggregateObject  
        (CreateObject<ConstantPositionMobilityModel> ());  
  
    Ptr<MobilityModel> mobility = nodes.Get(i)->  
        GetObject<ConstantPositionMobilityModel> ();  
  
    Vector pos = mobility->GetPosition();  
  
    pos.x = 1.0;  
    pos.y = 10.0;  
  
    mobility->SetPosition(pos);  
}
```

Camp, T. and Boleng, J. and Davies, V. A survey of mobility models for ad hoc network research, *Wireless Communications and Mobile Computing*, 2002.

Figures etc. in this presentation adapted from this paper.