

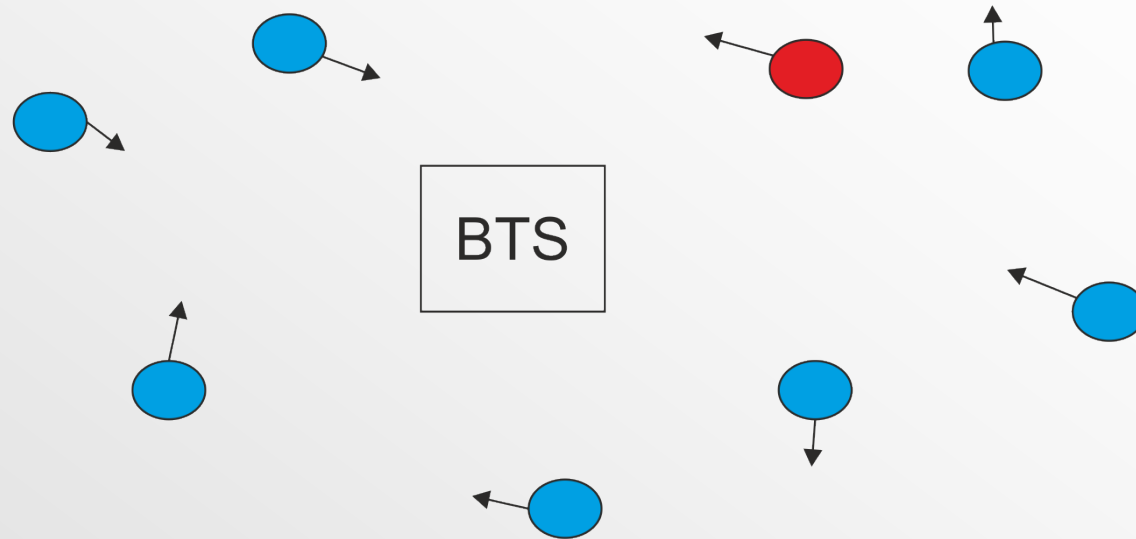
**TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY**

Kognitívne rádio

Peter Čarnakovič

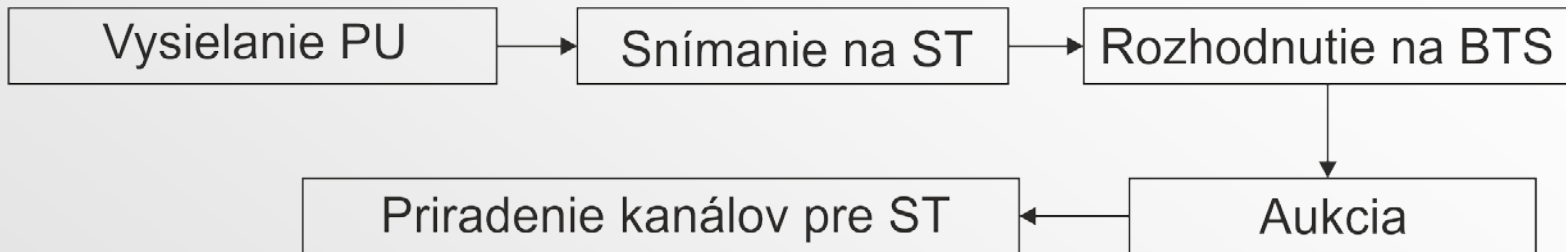
Vedúci práce: prof. Ing. Stanislav Marchevský, CSc
Konzultant: Ing. Lukáš Sendrei

- Vytvorte knižnice potrebné pre vytvorenie reálneho signálu v JAVE
- Upravte existujúci model tak, aby podporoval viackanálový prenos
- Pre daný model vytvorte 2 typy aukcií: sekvenčnú a súčasnú
- Jednotlivé typy aukcií a rozhodovacie metódy porovnajte pre vybrané parametre



- Primární uživatel
- Sekundární terminál

Model (2)



ST - Secondary Terminal (sekundárny terminál)

PU - Primary User (primárny používateľ)

BTS - Base Transceiver Station (základňová stanica)

- AnyLogic + JAVA

The screenshot displays the AnyLogic Professional interface. On the left, a project tree shows the structure of the simulation, including 'SpectrumSensingAndTrading3' and its sub-components like 'Main', 'SecondaryBTS', and 'SensitivityAnalysis_ser'. The main workspace shows a statechart diagram with several states: 'WAIT_FOR_SENSING_INFO', 'SENSING_DECISION', 'PREPARE_FOR_AUCTION', and 'AUCTION'. The 'SENSING_DECISION' state is currently selected, and its properties are shown in the right-hand pane. The 'Entry action' for this state is a Java code block that implements the logic for sensing decisions, including clearing decisions, iterating through channels, calculating risk, and updating the simulation state.

```
decisions.clear();
for(int i=0; i<get_Main().channels; i++) {
    int d = 0;
    risk = 1;
    for (STdecision st : global_sensing_info) {
        if (st.decision[i]) {
            d++;
        }
        risk = risk*st.Pd; //AND rule
        //risk = risk*(1-st.Pd); //OR rule
        //risk = 1-risk; //OR rule
    }
    risk=1-risk;
    //if(d==get_Main().secondaryTerminal.size()) //AND rule
    if(d>0) {
        decisions.add(true);
    } else {
        freeChannels++;
        decisions.add(false);
        if(get_Main().primaryUser.transmission.get(i)) {
            errors++;
        }
        total_risk += risk;
        get_Main().timeframe++;
    }
}

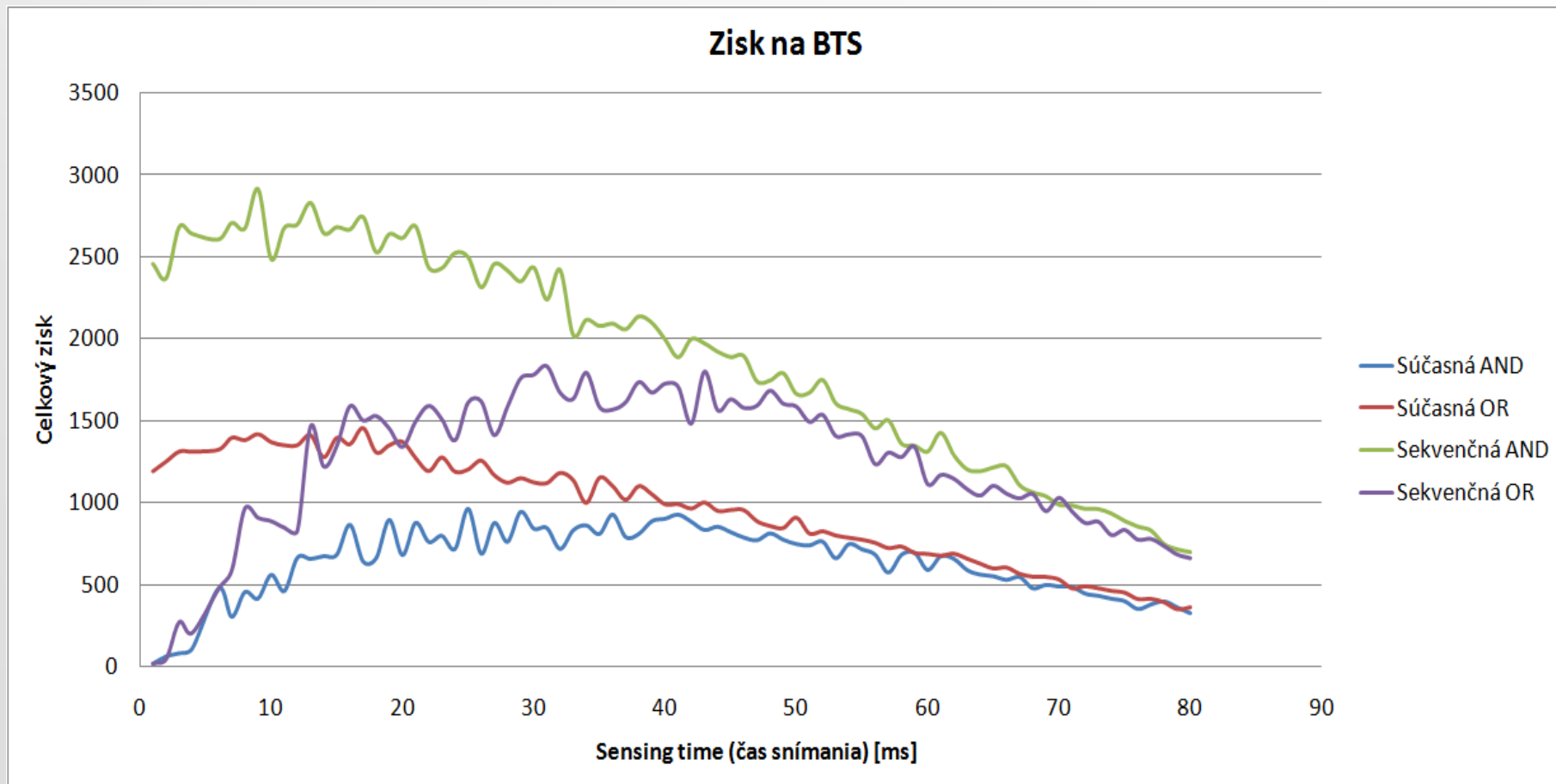
BTSdecision D = new BTSdecision(decisions, risk);
for(SecondaryTerminal ST : get_Main().secondaryTerminal) {
    send(D,ST);
}
```

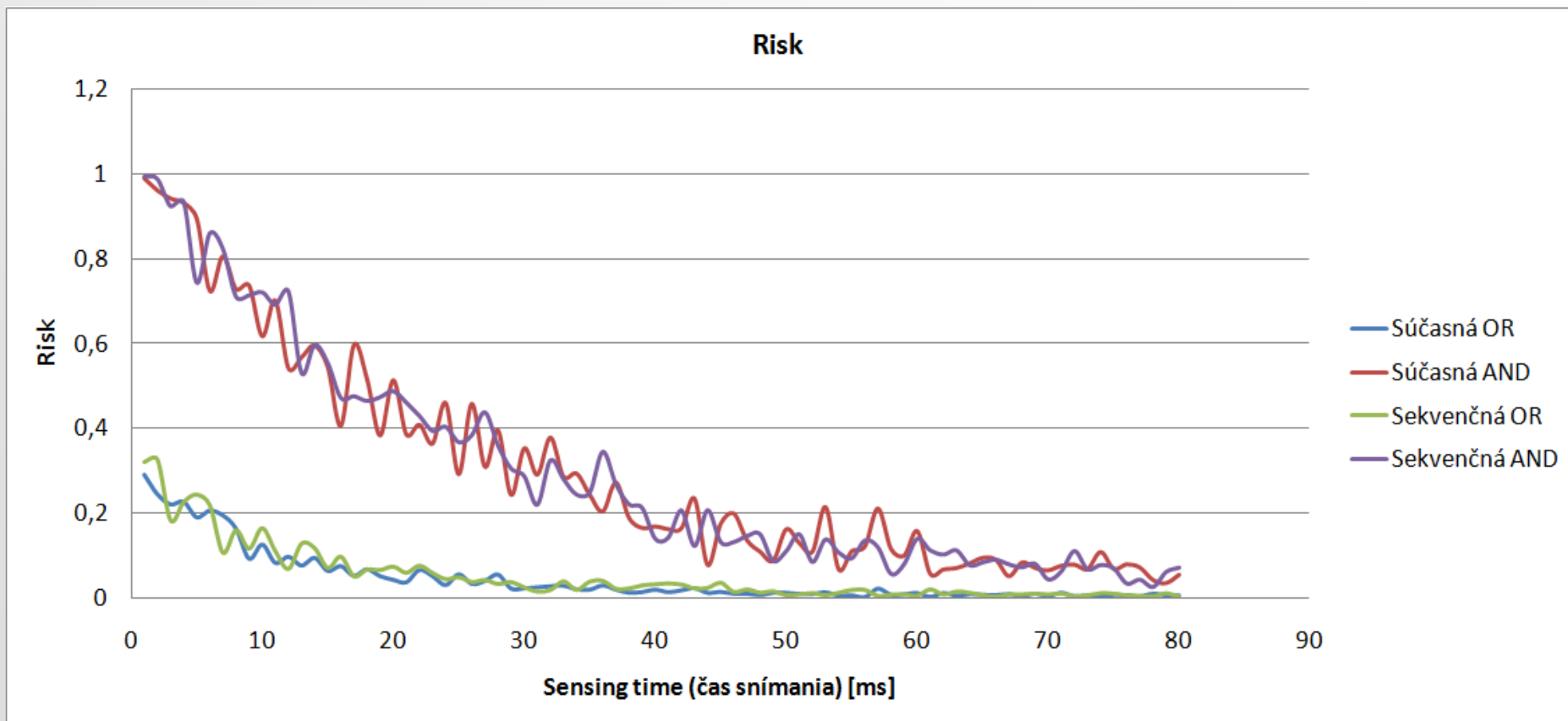
- Metódy OR a AND
- Rozhodnutie BTS či je daný kanál obsadený
- Metóda OR – pre rozhodnutie stačí 1 terminál, ktorý povie, že na danom kanále sa vysiela
- Metóda AND – všetky terminály musia súhlasiť, že na danom kanále sa vysiela

AND – logický súčin $1*1*0 = 0$

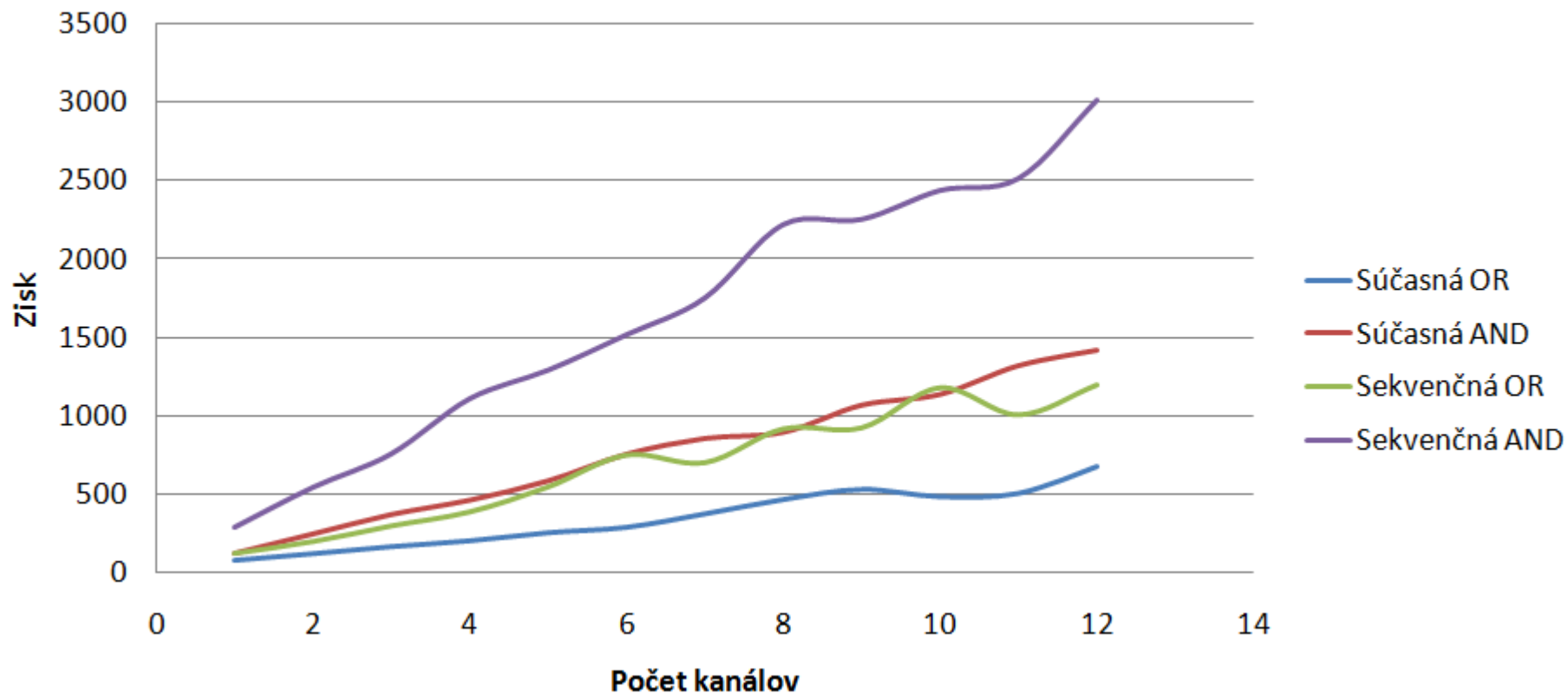
OR – logický súčet $1+1+1+0 = 1$

- 2 typy: súčasná a sekvenčná
- Aukcia prebieha na BTS
- Aukciu vyhráva terminál, ktorý ponúkne najvyššiu cenu za daný kanál
- Súčasná – naraz sa rozhodne, ktoré terminály získajú voľné kanály
- Sekvenčná – v jednom kole sa rozhoduje len pre jeden voľný kanál. V ďalšom kole sa rozhoduje už pre ďalší voľný kanál, ale terminál ktorému bol už priradený kanál sa nemôže zúčastniť na ďalších kolách





Zisk v závislosti od počtu kanálů



- Nasimulované hodnoty potvrdili naše očakávania
- Podľa výsledkov simulácií dostaneme najlepší výsledok, ak použijeme rozhodovaciu metódu OR, sekvenčnú aukciu a dĺžku snímania 25 ms
- Ďalšie možné vylepšenia: vylepšenie existujúceho modelu, aby sme sa čím viac priblížili k reálnym podmienkam – aplikovať reálny signál, použitie rôznych modulácií, šum, iný typ detektoru na snímanie atď.

Ďakujem za vašu pozornosť.