



МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ  
НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. ІГОРЯ СІКОРСЬКОГО»  
НАВЧАЛЬНО-НАУКОВИЙ ФІЗИКО-ТЕХНІЧНИЙ ІНСТИТУТ  
КАФЕДРА ІНФОРМАЦІЙНОЇ БЕЗПЕКИ

## **Кіберзахист об'єктів критичної інфраструктури**

### **Лабораторний практикум №3**

#### **Використання симулятора Сооја для моделювання об'єктів критичної інфраструктури**

Перевірив:

Войцеховський А. В.

Виконав:

студент I курсу

групи ФБ-41мп

Сахній Н. Р.

Київ 2024

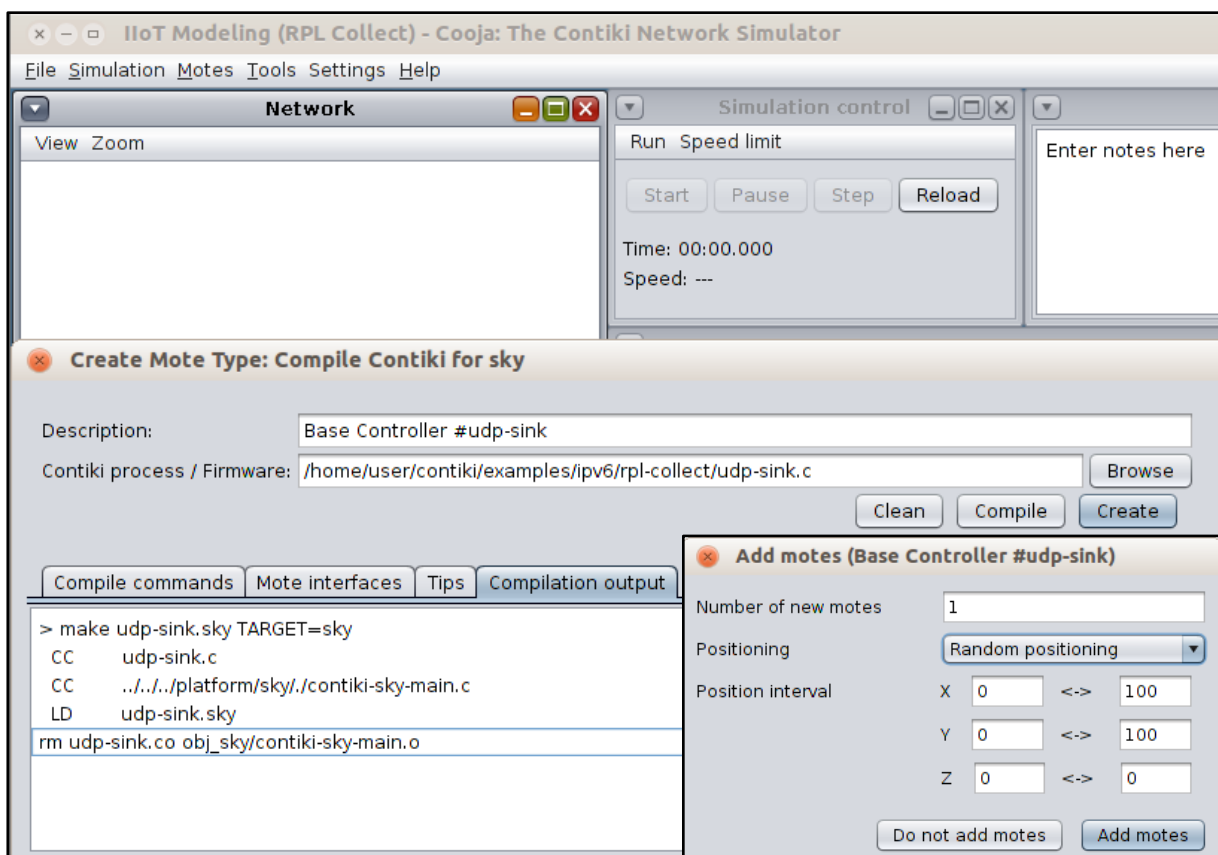
**Мета роботи:** Ознайомитись із можливостями симулятора Cooja для Contiki OS. Навчитись використовувати його для моделювання поведінки базових пристроїв промислового інтернету речей (IIoT).

**Завдання до виконання:**

**Варіант №5.** Дано 5 вузлів типу UDP-sender та 1 UDP-sink.

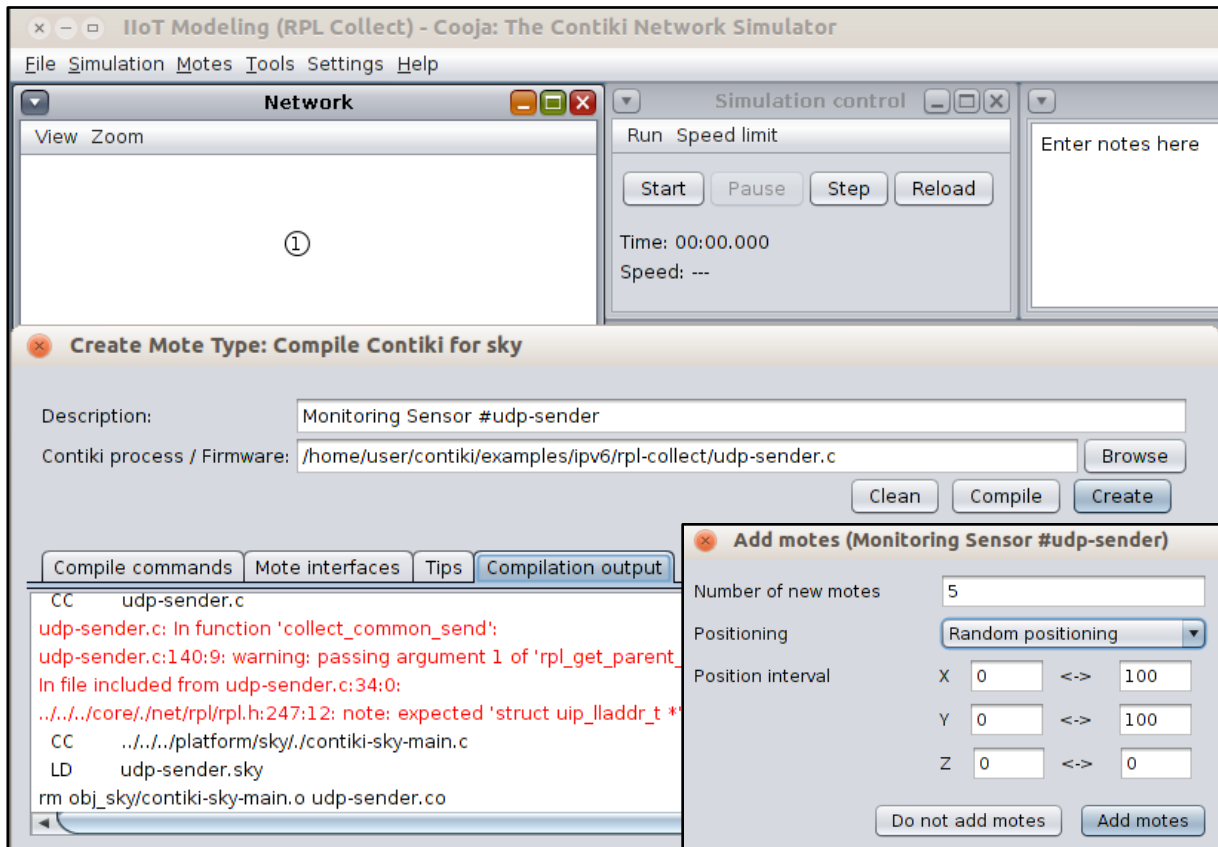
1. Обравши тип udp-sink.c, натиснути Compile. Повідомлення у вікні вида LD udp-sink.sky означає, що створився необхідний файл (це фактично прошивка для емульованого пристрою). Натиснути Create. Задати кількість пристроїв цього типу, Random positioning (для задання фізичного розташування), та Add mote. Вузол графічно відобразиться.

**Призначення контролера** – збирати дані для аналізу та прийняття рішень:



- Аналогічно додати декілька за варіантом пристроїв типу UDP-sender із /home/user/contiki/examples/ipv6/rpl-collect/udp-sender.c. Вони графічно відобразяться у відповідному віконці.

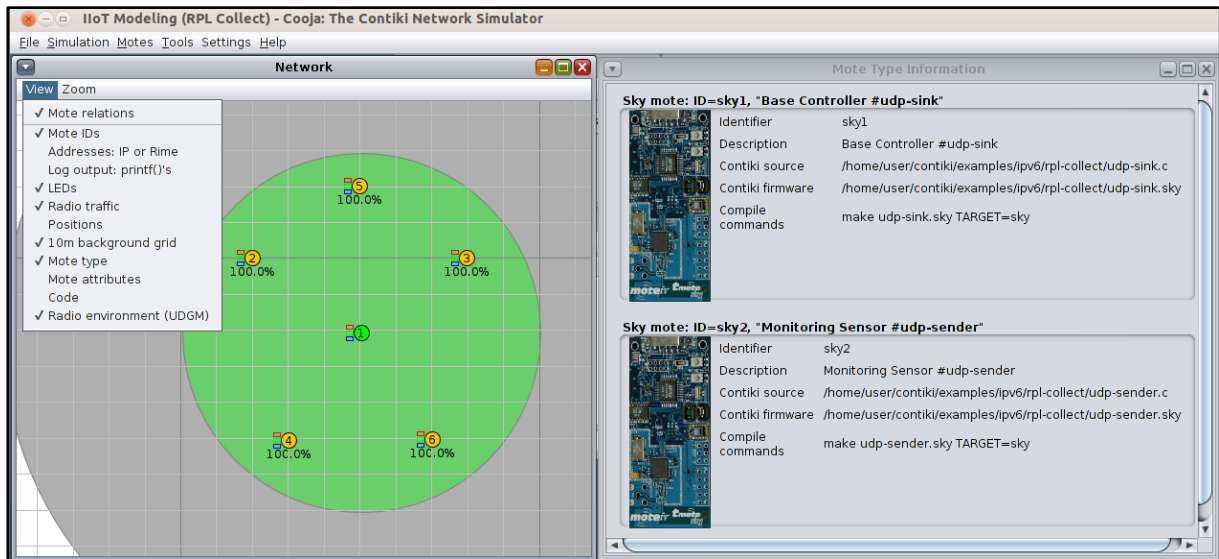
**Призначення датчиків** – відправляти значення технологічних показників:



- Для покращення візуалізації у вікні емульованої мережі Network вибрати у вкладці View опції Mote IDS, RadioTraffic, BackGroundGrid, MoteType, RadioEnvironment. Натискаючи на зображення пристрою у віконці можна побачити його радіус дії. Пристрої, які надсилають інформацію, повинні будуть знаходитись у радіусі дії приймача.

**Значення 100.0%** вказує на рівень успішності передачі даних або іншими словами силу сигналу між вузлами. Якщо якийсь із сенсорів вийде за межі

цієї зони, він не зможе передавати дані до контролера, і мережа втратить частину інформації, що не допустимо при здійсненні моніторингу:



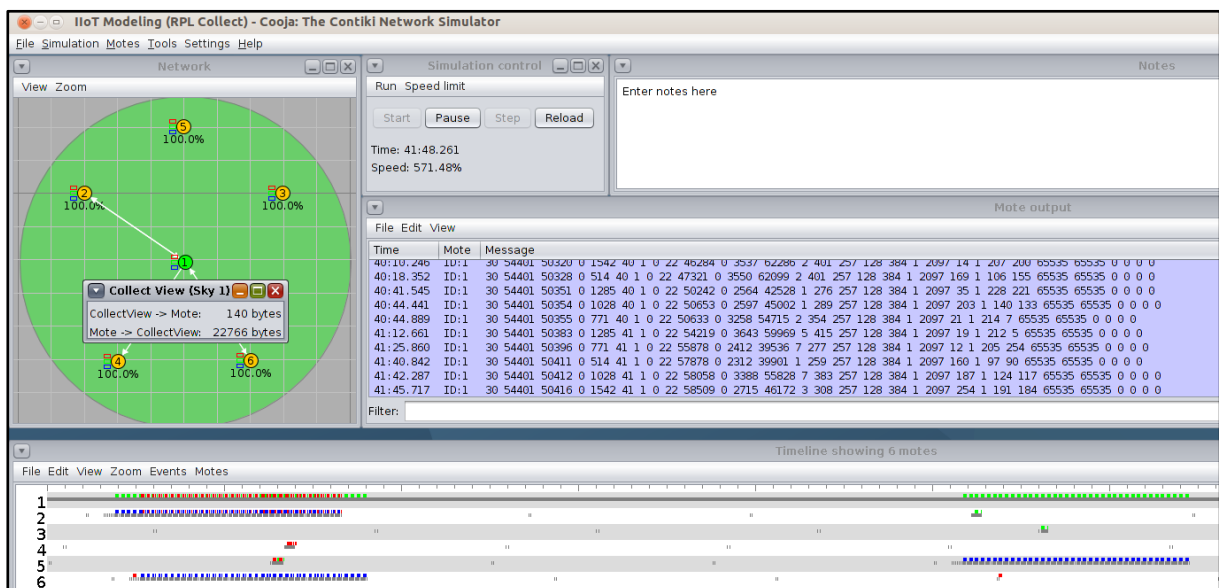
4. Тепер необхідно зібрати дані про поведінку мережі:

Tools → Collect View → Sky1

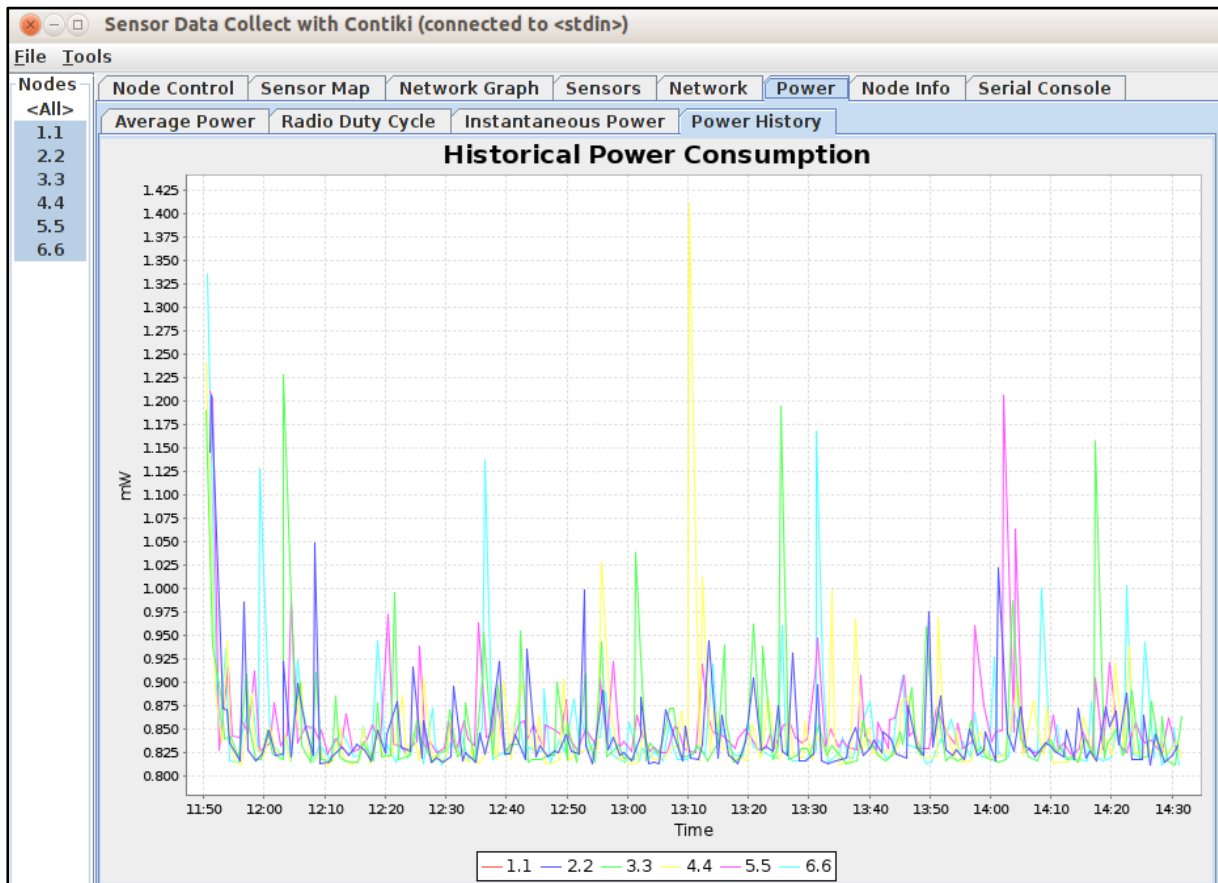
Simulation Control → Start

Start Collect | Send command to nodes

Зачекайте декілька хвилин, щоби накопичити дані.



5. Подивитись таблицю із даними про поведінку мережі у вкладці **Node Control** → **Node Info**. Звернути увагу на споживання електроенергії.



Далі виконати такі дії у інтерфейсі:

[Send stop to nodes](#)

[Stop Collect](#)

Звернути увагу на дані у стовпці Power таблиці даних.

Sensor Data Collect with Contiki (connected to <stdin>)															
File Tools															
Nodes <All> 1.1 2.2 3.3 4.4 5.5 6.6	Node Control		Sensor Map		Network Graph		Sensors		Network		Power		Node Info		Ser
	N...	Received	Dups	Lost	Hops	Rtmetric	ETX	Churn	Beacon Interval	Reboots	CPU ...	LPM ...	Listen...		
	1.1	0	0	0	0.000	0.000	0.000	0		0	0.000	0.000	0.000		
	2.2	244	0	0	1.000	393.480	16.000	0	33 min, 26 sec	0	0.303	0.154	0.378		
	3.3	244	0	0	1.000	393.557	16.000	0	33 min, 21 sec	0	0.303	0.154	0.398		
	4.4	244	0	0	1.000	393.361	16.000	0	33 min, 25 sec	0	0.302	0.154	0.380		
	5.5	244	0	0	1.000	393.484	16.000	0	33 min, 27 sec	0	0.313	0.154	0.378		
	6.6	244	0	0	1.000	393.443	16.000	0	33 min, 22 sec	0	0.303	0.154	0.378		
	Avg	244.000	0.000	0.000	1.000	393.465	16.000	0.000	33 min, 24 sec	0.000	0.305	0.154	0.382		

al Console							
Transmit...	Power	On-time	Listen Duty Cycle	Transmit Duty Cycle	Avg Inter-packet Time	Min Inter...	Max Inter...
0.000	0.000		0.000	0.000			
0.011	0.846	48 min, 35 sec	0.631	0.021	0 min, 59 sec	0 min, 0...	1 min, 56...
0.013	0.868	48 min, 39 sec	0.663	0.024	0 min, 59 sec		1 min, 58...
0.010	0.847	50 min, 08 sec	0.634	0.020	0 min, 59 sec	0 min, 1...	1 min, 53...
0.012	0.857	49 min, 41 sec	0.630	0.023	0 min, 59 sec	0 min, 0...	1 min, 52...
0.011	0.846	49 min, 28 sec	0.630	0.020	0 min, 59 sec	0 min, 0...	1 min, 57...
0.011	0.853	49 min, 18 sec	0.637	0.021	0 min, 59 sec	0 min, ...	1 min, 5...

За наведеними даними в таблиці “Node Info” можна зафіксувати наступне:

#### ❖ Енергоспоживання:

- Середнє споживання електроенергії: CPU Power: 0.305, LPM (Low Power Mode): 0.154, Listen Power: 0.380, Transmit Power: 0.015.
- Загальне енергоспоживання коливається навколо 0.850 мВт для кожного вузла, що свідчить про низьке енергоспоживання системи під час роботи. Це вказує на ефективність використання енергії, що є важливим для розгорнутих мереж датчиків, де ресурси обмежені.

#### ❖ Робочий цикл:

- “Listen Duty Cycle” знаходиться на рівні 0.633 для більшості вузлів, що свідчить про те, що датчики значну частину часу витрачають на очікування даних або подій, що є стандартною поведінкою для подібних пристроїв.
- “Transmit Duty Cycle” вкрай низький – 0.028, що свідчить про те, що більшість часу вузли не передають дані, але вони все ж регулярно пересилають пакети.

#### ❖ Передача даних:

- Усі вузли прийняли 244 повідомлення без втрат даних (Dups = 0, Lost = 0), що свідчить про стабільність мережі. При цьому кожен

датчик має відстань у 1 “hop” (перехід) до контролера, що є ознакою прямого зв'язку без ретрансляції через інші вузли.

❖ **Інтервал між пакетами:**

- “Avg Inter-packet Time” становить приблизно 1 хвилину (59 секунд), мінімальний інтервал – 8 секунд, а максимальний – близько 2 хвилин. Це може означати, що в мережі не відбувається великої кількості одночасних передач, і вузли обмінюються даними досить рідко, що знижує навантаження на мережу.

**Загальний висновок:** Мережа стабільно функціонує, не відбувається втрат даних, а енергоспоживання вузлів залишається на низькому рівні. Це говорить про те, що система ефективно управляє передачею даних та економією енергії, що є критично важливим для мереж промислового інтернету речей (IIoT).

**6. Відповіді на контрольні запитання.**

▪ **Призначення та основні переваги Contiki OS.**

Contiki OS – це операційна система для мережних систем з обмеженою пам'яттю. Акцент у використанні цієї ОС робиться на пристроях Інтернету речей (IIoT), які використовують безпроводний зв'язок. На даний момент Contiki OS використовувалось для підтримки систем забезпечення роботи розумного міста. Зокрема, вуличного освітлення та звукового вуличного освітлення, радіаційного моніторингу, сигналізації. Перевагою цієї ОС є те, що вона є системою з відкритим кодом, що випускається під ліцензією BSD, а також побудована на модульній архітектурі, що дозволяє додавати компоненти під конкретні завдання.

▪ **Види пристроїв серед прикладів у /home/user/contiki/examples/.**

Існують різні види пристроїв, які можуть бути корисні для симуляцій в Contiki OS. Нижче наведено перелік основних прикладів:

- \* **antelope** – бібліотека для управління реляційними базами даних у сенсорних мережах (системі технологічних датчиків).
- \* **cc2530dk/cc2538dk** – приклади для роботи з наборами розробки на базі чіпів CC2530/CC2538 від компанії Texas Instruments, які використовуються для бездротових мереж (наприклад, Zigbee).
- \* **collect** – приклад для роботи зі збиранням даних у мережі сенсорів.
- \* **eeprom-test** – тестування пам'яті EEPROM на сенсорів.
- \* **er-rest-example/rest-example** – приклади використання RESTful веб-сервісів для взаємодії з сенсорними пристроями.
- \* **sky** – приклади для роботи з платформою сенсорів TelosB.
- \* **udp-stream** – приклад передачі поточкових даних через UDP.
- \* **z1/z1sp** – приклади для роботи з платформою Zolertia Z1, яка широко використовується в сенсорних мережах.

Крім фізичних пристроїв, також присутні приклади сервісних аплікацій:

- \* **email**: приклад надсилання електронних листів із пристроїв IoT.
- \* **ftp**: приклад реалізації протоколу FTP для віддаленого завантаження або обміну файлів між пристроями в мережах IoT.
- \* **webserver**: приклади простих веб-серверів, які дозволяють керувати або отримувати дані з пристроїв через веб-інтерфейс.

Ці приклади призначені для роботи з різними платформами, що дозволяє розробникам проводити тести на різних типах пристроїв у мережах IoT.



- **Яким чином здійснюється моделювання мережі у Сооја?**

- Яким чином задається поведінка пристрою?**

- Моделювання мережі здійснюється за допомогою емулятора мережі, який імітує поведінку вузлів Contiki OS. Він використовує моделі протоколів і обладнання для імітації передачі даних між пристроями.

- Поведінка пристрою задається за допомогою файлу програми, який компілюється та завантажується на пристрій (udp-sink.c, upd-sender.c).

- **Чому важливо задавати фізичне розташування змодельованих пристроїв, а не лише їх внутрішні налаштування?**

- Досить важливо – це фізично розташувати пристрої так, щоб вони знаходилися в зоні досяжності один одного і могли взаємодіяти між собою, а саме обмінюватися інформацією.

- **Які задачі безпеки можна вирішувати із використанням Сооја?**

- \* **Аналіз атак на мережі IoT:** Сооја може використовуватися для моделювання атак на IoT-мережі, таких як MitM чи DDoS.
  - \* **Моделювання безпечної маршрутизації:** Сооја дозволяє досліджувати і перевіряти різні алгоритми маршрутизації з урахуванням факторів безпеки, таких як захист від спрямування мережевого трафіку через небезпечні маршрути.
  - \* **Оцінка енергетичних витрат на безпеку:** Сооја може допомогти змодельовати, як різні підходи до забезпечення безпеки (наприклад, використання захищених протоколів) впливають на споживання електроенергії пристроями в мережі.