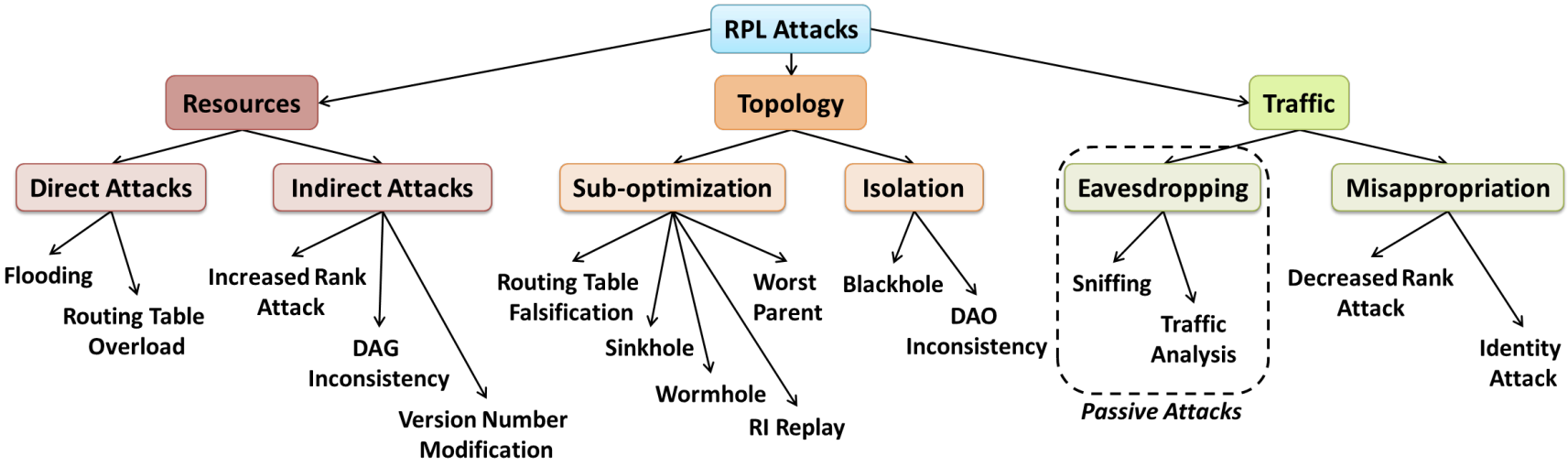
Blackhole Attack Tutorial

Topology Attacks

- Topology attacks are one category of security attacks on the RPL protocol, as shown below
  - Their purpose is to disturb the topology building mechanisms of the network, for instance in order to cause the isolation of one or more nodes



Source: https://hal.inria.fr/hal-01207859/document

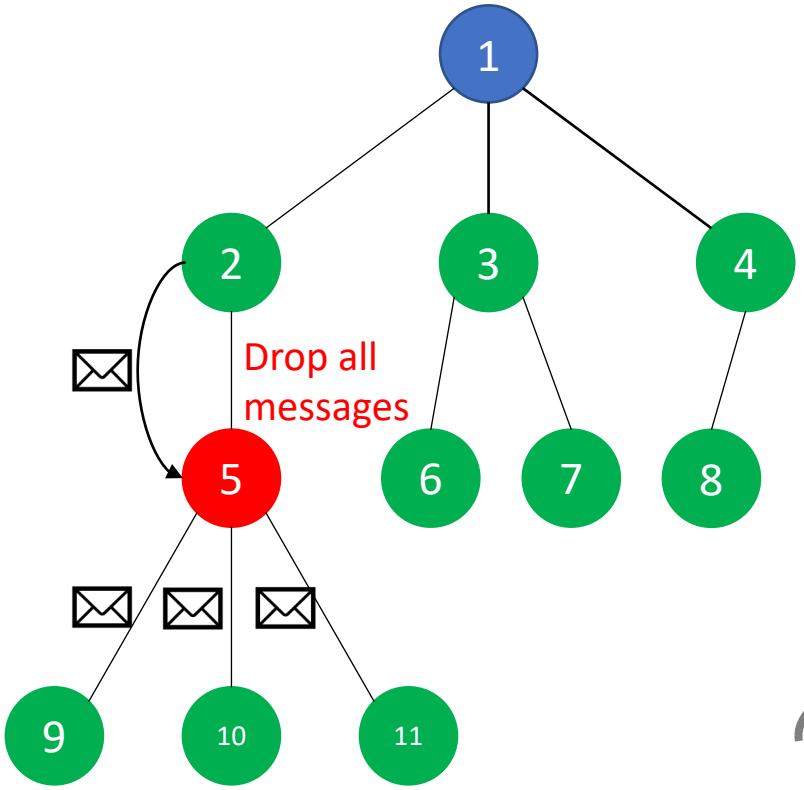
Topology Attacks (cont.)

- Topology attacks can be divided in two classes
  - Sub-optimization, meaning that the network under attack will converge to a non-optimal form, thus inducing poor performance
  - Isolation, means to separate a node or subset of nodes and cut them from the rest of the network (hence also from the root node)
- Blackhole attacks are an example in the “Isolation” class that we shall address in more detail next

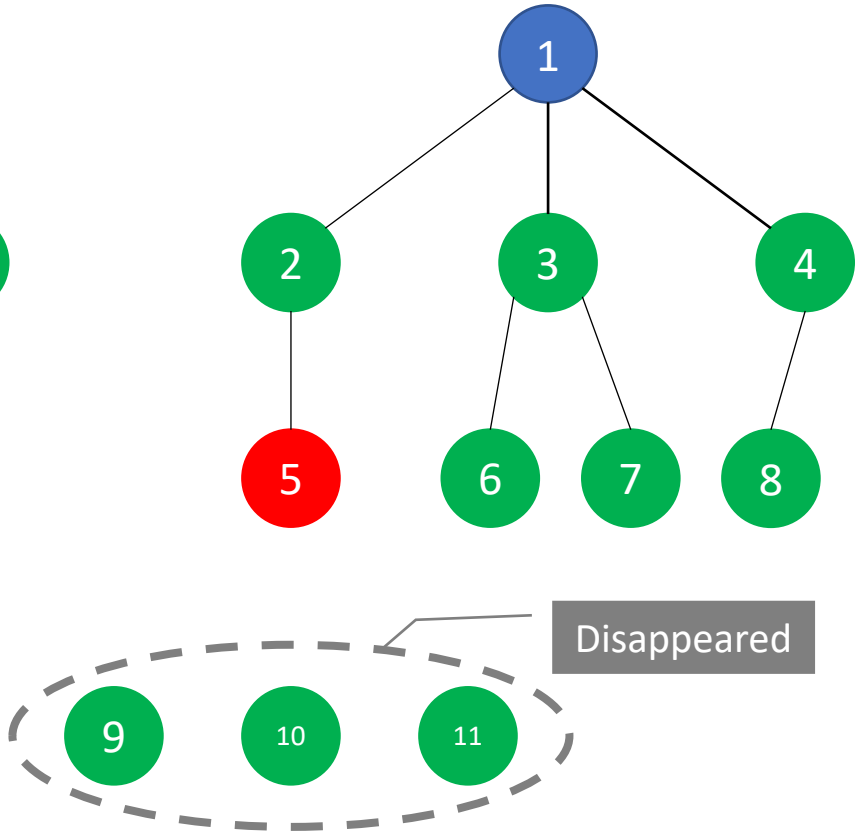
Blackhole Attacks

- Blackhole attacks aim to drop all the packets that the malicious node is supposed to forward
  - In a sense, this attack can also be seen as a partial denial-of-service attack
- If this attack is combined with a sinkhole attack, it can be very damaging, by causing the loss of the whole deflected traffic
- If the position of the malicious node is well chosen, it can isolate several nodes from the network
- The selective forwarding attack (a.k.a. gray hole) is a variant of this type of attack

Blackhole Attacks (cont.)



Initial network topology



Final network topology

Blackhole Attack Simulation

Blackhole Attack Simulation

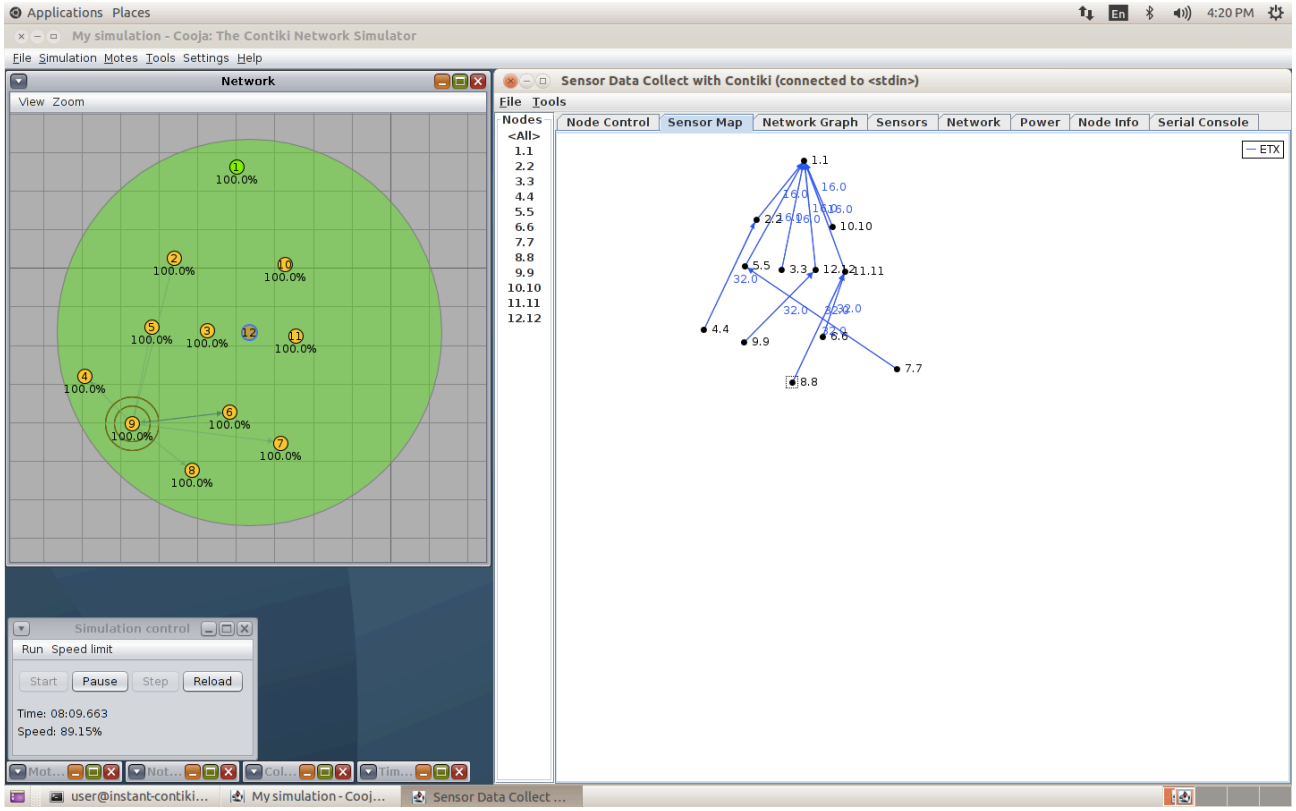
- Open the desired simulation in Cooja by selecting the corresponding scenario via the IoTrain-Sim interface
  - We recommend that you first select the “Reference Scenario Simulation” entry to view the reference scenario
- Alternatively, the simulations can be opened manually as follows
  - In Cooja, select the menu File > Open simulation > Browse…
  - Go to the folder “iotrain-sim/database/security_training/blackhole_attack/simulation/”
  - Select “blackhole_attack-reference.csc” for the reference scenario, and click “Open”

Blackhole Attack Simulation (cont.)

- Simulation and data collection procedure
  1. In the CollectView window, click on the “Start Collect” button, then click on the “Send command to nodes” button
  2. In the Simulation control window of Cooja, click on the “Start” button to begin the simulation
  3. Wait for at least two minutes of simulation time
  4. Back in the CollectView window, go to the Sensor map tab and see the network topology for the scenario
- Follow the same procedure to perform the attack simulation and compare the results
  - The attack scenario can be opened via the menu “Blackhole Attack Simulation” in IoTrain-Sim, or directly in Cooja via the file “blackhole_attack-simulation.csc”
  - You may need to wait for more than five minutes of simulation time to get statistics for all the nodes

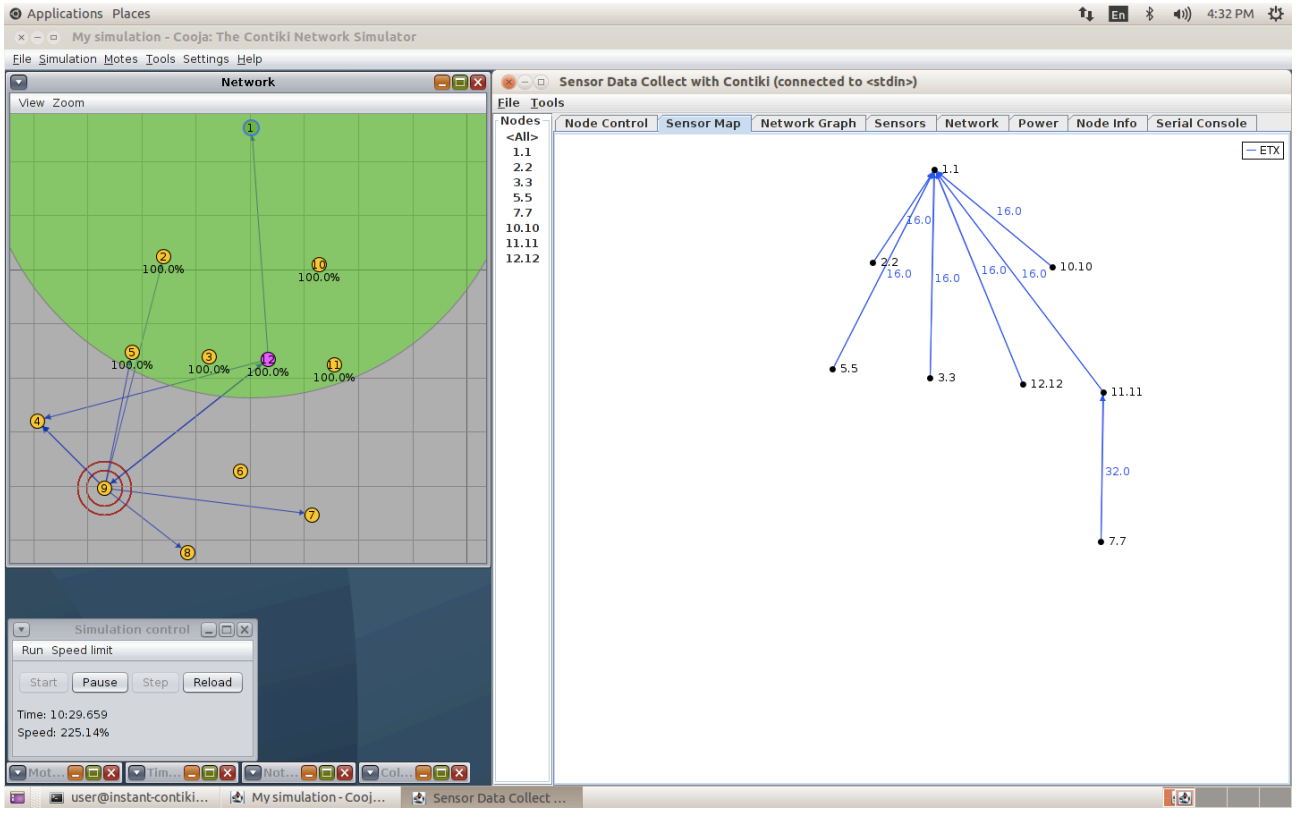
Reference Scenario and Results

- Node 1 (green color) is a SINK node that acts as a border router; the other nodes are sender nodes that act as normal sensors
- Notice that all the nodes are in the range of node 12, but nodes 4 through 9 are not in the range of node 1



Attack Scenario and Results

- Node 1 and the nodes in yellow color have the same roles as before
- Node 12 became a malicious node performing a blackhole attack



Discussion

- Reference scenario
  - All the nodes are included in the network topology shown in Sensor map panel
    - Notice that nodes can be moved around with the mouse in the Sensor map panel to make the topology easier to see
  - The nodes can automatically determine and select the best path to the root node
- Attack simulation
  - The blackhole effect of node 12 caused nodes 4, 6, 8 and 9, which previously connected through it to the network, to disappear from the topology
  - These nodes thus become disconnected, without any path for sending data to the root node

Blackhole Attack Implementation

Implementation Overview

- To implement the blackhole attack, some changes are necessary to the normal source code for the RPL implementation in Contiki
- The file to be modified is located in the directory “contiki/core/net/ipv6/”
  - uip6.c, which contains the implementation of the uIP TCP/IPv6 stack for Contiki

Changes to uip6.c

- The file “uip6.c” includes code that forwards packets not intended for the current node to their destination
- To implement the blackhole attack, the malicious node can drop all such packets
  - This will cause the affected nodes to stop receiving traffic from nodes they are not directly connected to

```
#endif /* UIP_CONF_IPV6_RPL */

    UIP_IP_BUF->ttl = UIP_IP_BUF->ttl - 1;
    PRINTF("Forwarding packet to ");
    PRINT6ADDR(&UIP_IP_BUF->destipaddr);
    PRINTF("\n");
    UIP_STAT(++uip_stat.ip.drop);                    //change from UIP_STAT(++uip_stat.ip.forwarded);
    goto drop;                                       //change from goto send;
  } else {
    if((uip_is_addr_link_local(&UIP_IP_BUF->srcipaddr)) &&
       (!uip_is_addr_unspecified(&UIP_IP_BUF->srcipaddr)) &&
```

Blackhole attack implementation via dropping all the packets that the malicious node is supposed to forward

Exercises

- After making the suggested modifications in a copy of the Contiki source code, compile the files and assign the resulting malicious firmware to one of the motes in the reference scenario
- We suggest you use node 12 first as malicious one, as in our example, then change the malicious node to another one and see how the results change
  - You can also use multiple malicious nodes and compare the simulation results