# Scheduling Intrusion Detection Systems in Resource-Bounded Cyber-Physical Systems

Waseem Abbas[1], Aron Laszka[2], Yevgeniy Vorobeychik[1], Xenofon Koutsoukos[1]

[1] Institute for Software Integrated Systems, Vanderbilt University
[2] Electrical Engineering and Computer Science Department, UC Berkeley


# Securing Cyber-Physical Systems

- Securing cyber-physical systems is challenging
  - long lifetime
  - difficult software updates
  - resource and timing constraints
  - ...

→ Practically impossible to prevent all attacks

- To mitigate losses arising from successful attacks, operators need to be able to **detect attacks**
  - detection enables reacting in time and preventing substantial losses

# Examples of Stealthy Attacks

- Maroochy Shire incident
  - disgruntled ex-employee issued radio commands to SCADA sewage equipment
  - on **at least 46 occasions** from February 28 to April 23, 2000
  - caused 800,000 liters of raw sewage to spill out into local parks and rivers

- Stuxnet worm
  - targeted Iranian uranium enrichment facilities
  - **subtly increased** the pressure on spinning centrifuges, while showing the control room that everything was normal
  - reportedly ruined one-fifth of Iran's nuclear centrifuges

# Intrusion Detection System (IDS)

- Monitors a system or network for malicious activity
  - network-based IDS: monitors traffic passing through to an entire subnet
  - host-based IDS: runs on and monitors a single system
- For example,
  - by monitoring file system objects for modifications
  - by detecting suspicious system call sequences
- Protecting the IDS
  - attackers may try to disable the IDS before an alarm is raised
    → IDS needs to be running in order to detect the attack
  - however, an effective IDS can be resource intensive

# IDS for Cyber-Physical Systems

- Challenges
  - low performance devices ⟷ IDS can be resource intensive
  - battery powered devices ⟷ long system lifetime

→ IDS cannot be running **continuously**

- Scheduling problem: *When to run the IDS?*
  - deterministic schedule
    ⟷ attacker will launch its attack when the IDS is not running
  - naïve randomization: uniform random
    ⟷ attacker will target the points that will result in maximum losses

→ schedule must be **tailored to the physical system**

# Scheduling Intrusion Detection Systems for Sensors in Water-Distribution Networks

# Leakages in Water-Distribution Networks

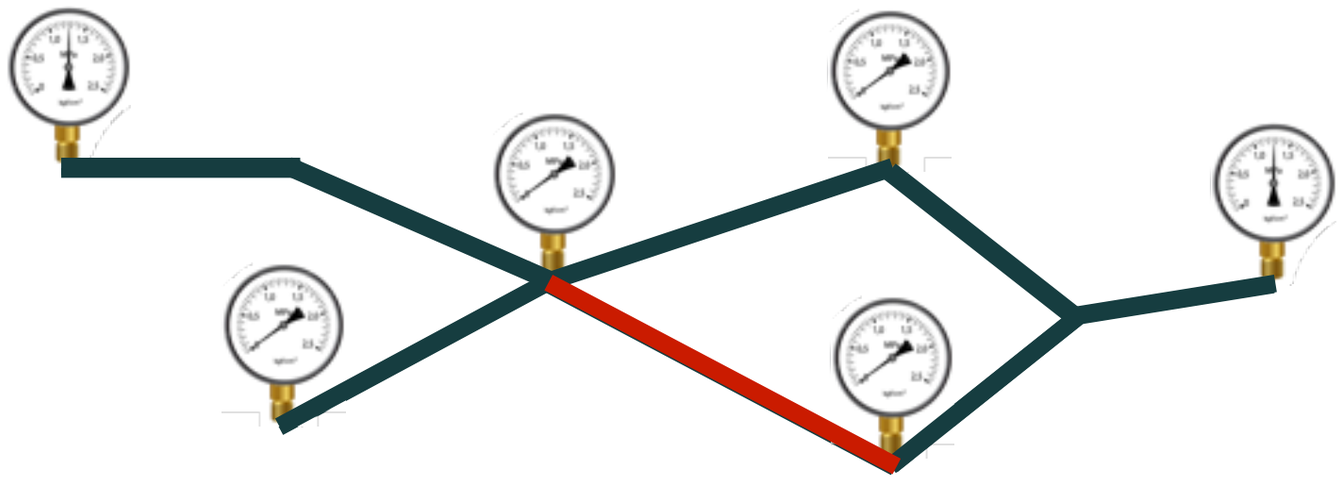
- Leakages can cause
  - significant economic losses
  - extra costs for final consumers
  - third-party damage and health risks
  - …

*“6 billion gallons of water per day may be wasted in the U.S.”*
(Center for Neighborhood Technology, 2013)

“*worldwide cost of physical losses is over $8 billion*”
(World Bank, 2006)

# Monitoring Water-Distribution Networks

- Pressure sensors can detect nearby events, such as leaks and pipe bursts



- An attacker might compromise a subset of sensors and change their observations
  - both false alarms and undetected leaks can result in economic losses
- Host-based IDS may be deployed to detect cyber-attacks
  - however, battery-powered sensor devices pose a scheduling problem

# Water-Distribution Network Model

- **Network**: graph $G(V, E)$
  - nodes $V$ correspond to junctions
  - links $E$ correspond to pipes
- **Sensors**: node subset $S \subseteq V$
- **Detection**:
  a sensor can detect a leakage at a pipe (i.e., link) if the distance between the sensor and the farther endpoint of the link is at most $D$
- **Time**: divided into $T$ time-slots, denoted $1, \ldots, T$
- **Battery**: each sensor can run IDS for at most $B$ time-slots

# Security Problem

- **Schedule**: for each time-slot $t$, the set $S_t$ of sensors running IDS

$$\forall s \in S : \sum_{t=1}^{T} 1_{\{s \in S_t\}} \leq B$$

- **Randomization**:
  sets are activated in a random order to prevent an attacker from predicting which sensors are running IDS in a given time-slot

- Attacker
  - chooses a link $\ell$ and changes the leakage report by compromising the sensors $A(\ell)$ that can detect link $\ell$
  - minimizes the probability of detection

| | Worst-case attacker | Random attacker |
|---|---|---|
| = | $\min_{\ell \in E} \sum_{t=1}^{T} 1_{\{A(\ell) \cap S_t \neq \emptyset\}}$ | $\frac{1}{\lvert E \rvert} \sum_{\ell \in E} \sum_{t=1}^{T} 1_{\{A(\ell) \cap S_t \neq \emptyset\}}$ |

- **Optimal schedule**: maximizes the probability of detection by IDS

# Computational Complexity

**Theorem 1:** Given an instance of our model, determining whether there exists a schedule that detects every attack with probability one is an NP-hard problem.

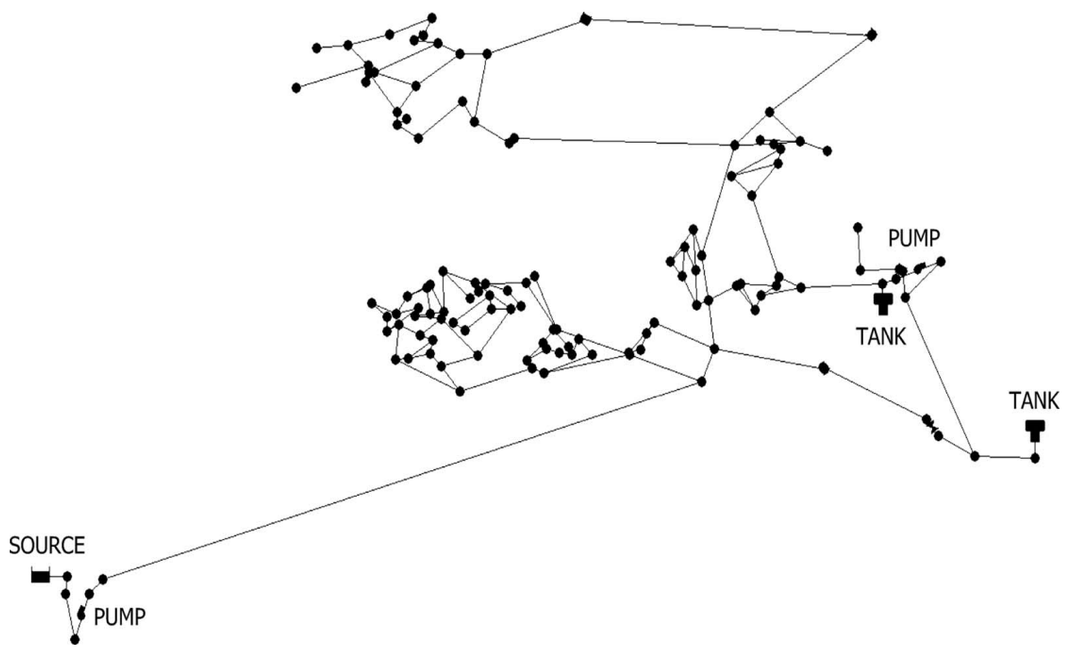
- We prove computational complexity for the special case $D = 2, B = 1$, and $T = 2$
- We propose **heuristic** algorithms for finding schedules against both **worst-case** and **random** attackers

# Heuristics for Worst-Case Attackers

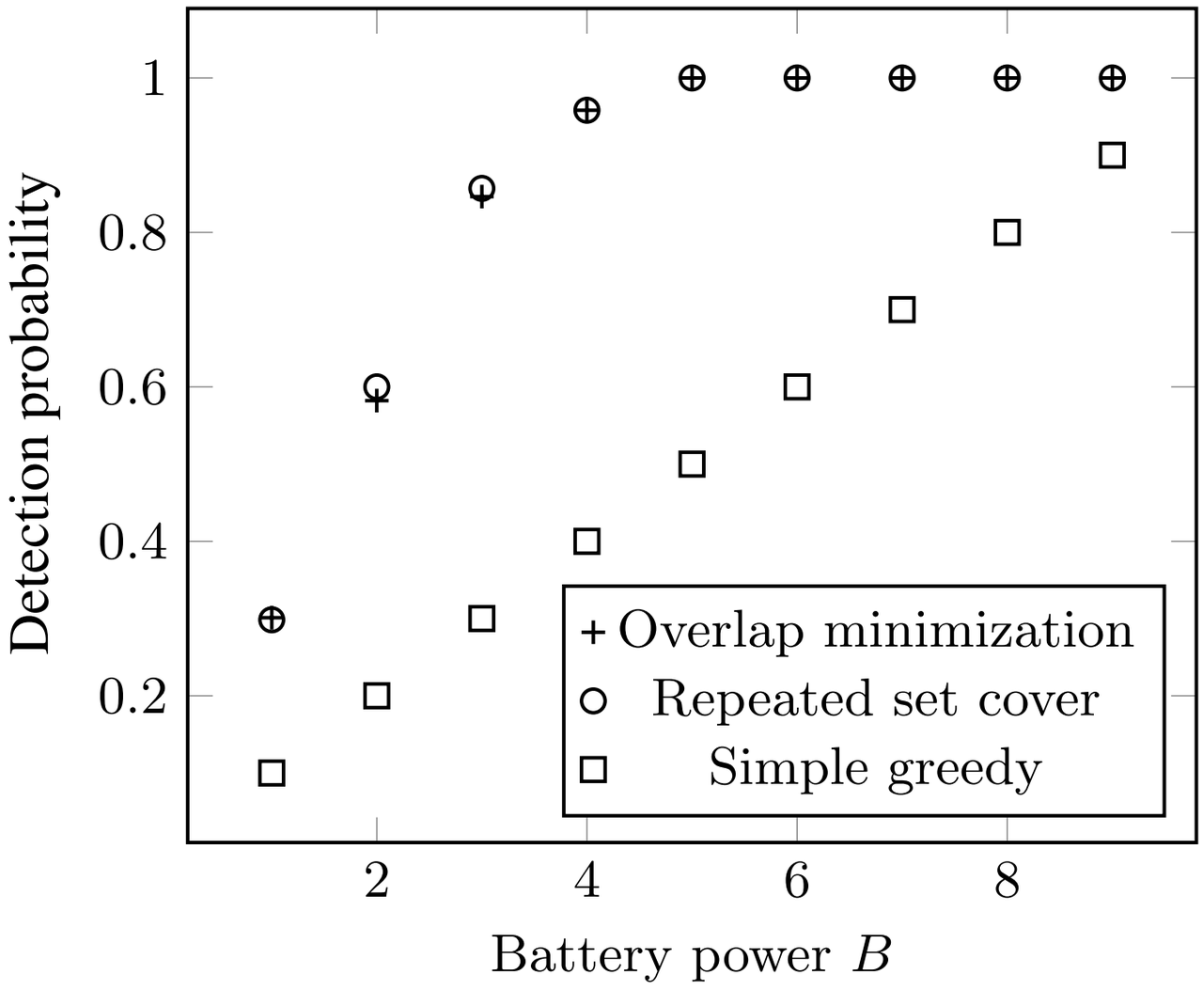
- Simple greedy
  - start with an empty schedule
  - assign sensors to the sets $S_t$ iteratively, always choosing a feasible combination that maximizes detection probability
- Overlap minimization
  - assign sensors to the sets $S_t$ iteratively, always choosing a feasible combination that minimizes overlap between sensors
  - i.e., avoid covering links that are already covered in a time-slot
- Repeated set cover
  - iterate over the time-slots, finding a minimal set cover for each time-slot
  - if there is no covering set of sensors left, maximize coverage using all the sensors

# Numerical Evaluation

- Random graphs
  - **geometric**: nodes are drawn from a unit square uniformly at random, and two nodes are connected if their distance is less than 0.15
  - **Barabási-Albert (BA)**: starting from a clique of 2 nodes, each additional node is connected to 2 existing nodes using preferential attachment
- For both types, we generated 1000 graphs, each graph having 100 nodes
- Real water-distribution network
  - 126 nodes and 168 pipes
  - from Ostfeld et al.: "*The Battle of the Water Sensor Networks (BWSN): A Design Challenge for Engineers and Algorithms*"

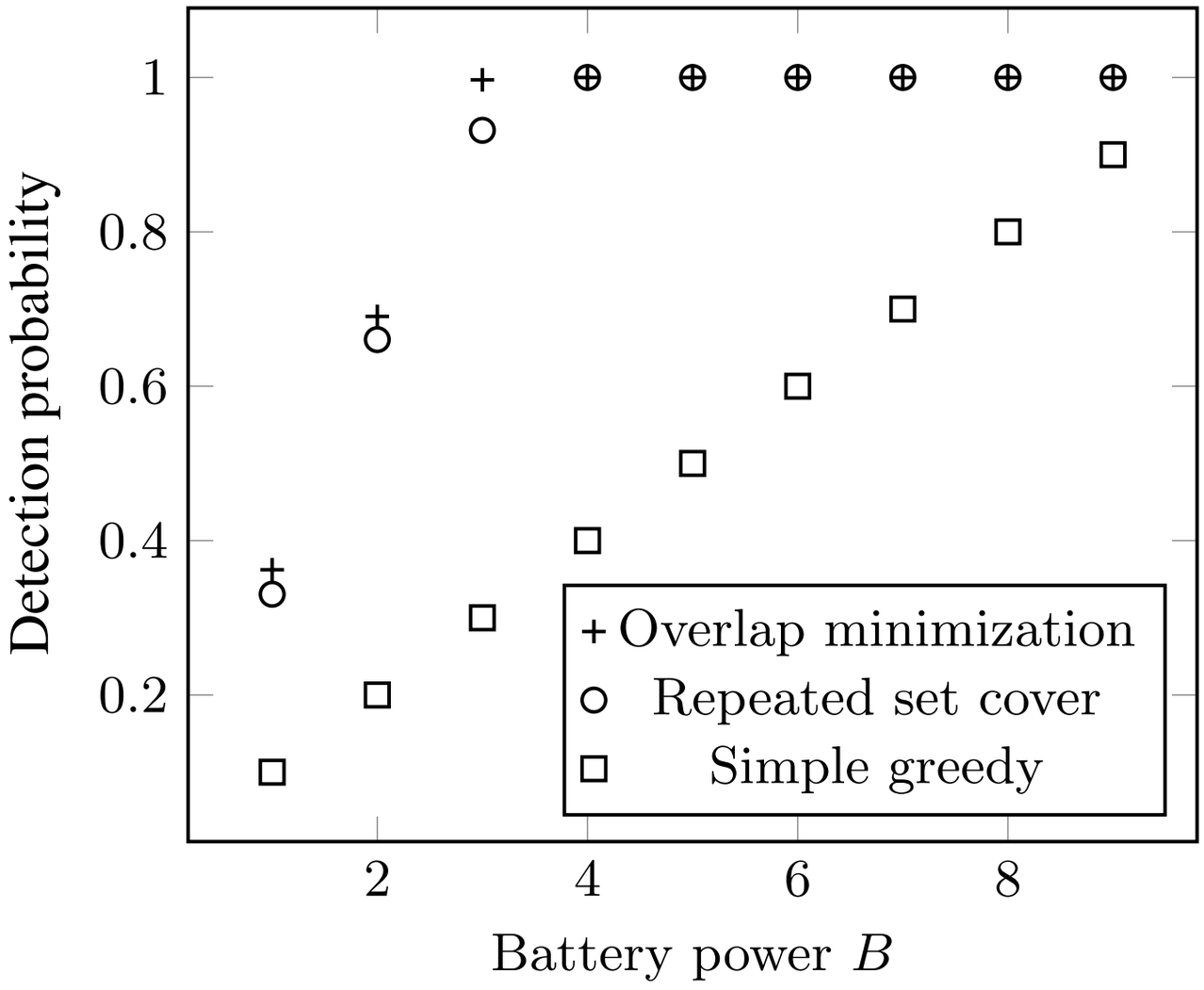


# Numerical Results / Geometric Graphs



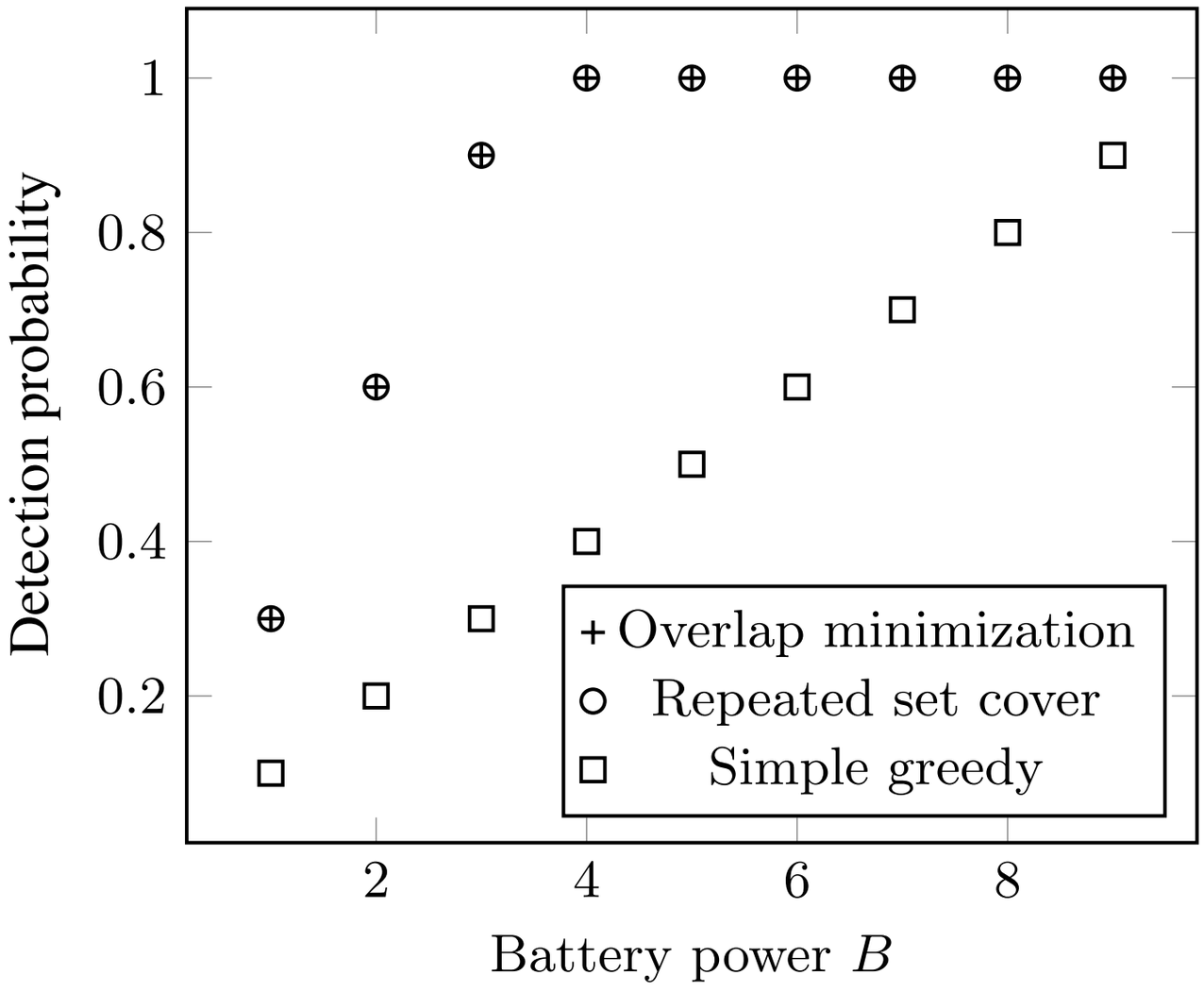

$S = V$, $D = 2$, and $T = 10$

# Numerical Results / B-A Graphs




$S = V$, $D = 2$, and $T = 10$

# Numerical Results / Real Water Network




$S = V$, $D = 2$, and $T = 10$

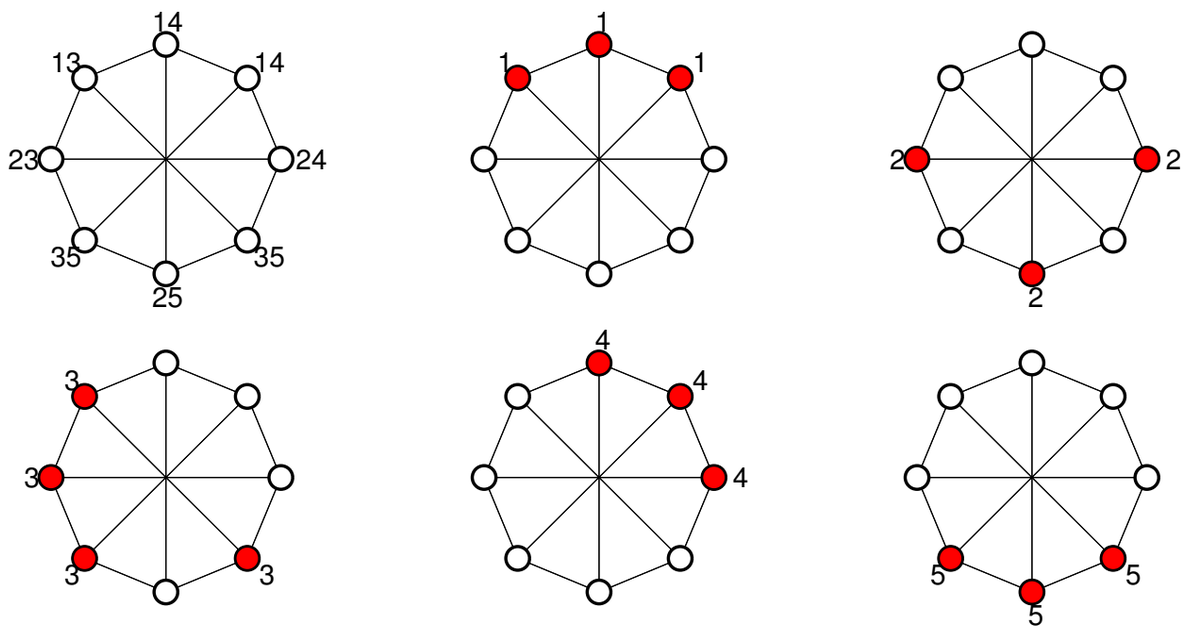
# Heuristics for Random Attackers

- We constrain the detection distance $D$ to be $2$
- Sufficient condition for perfect detection
  - if every $S_t$ is a dominating set, then every attack is detected
    - dominating set:
      every node is either an element of the set or one of its neighbors is
- Heuristic approach:
  *find a maximum set of dominating sets*

# Finding Dominating Sets

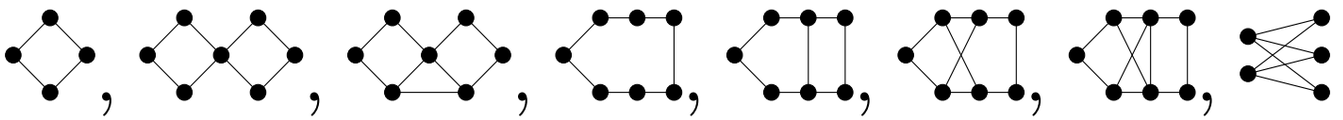
- Disjoint dominating sets
  - partition the node set into pairwise disjoint dominating sets
  - domatic number $\gamma$: maximum number of disjoint dominating sets
  - achievable lifetime $T = \gamma B$
- Non-disjoint dominating sets
  - we can achieve longer lifetime if the sets are not disjoint

$B = 2$



# Finding Non-Disjoint Dominating Sets

- $(r, s)$-configuration: assignment of $s$ distinct labels to each node from a set of labels $\{1, \ldots, r\}$, such that for every label $l$ and every node $v$, label $l$ is assigned to node $v$ or one of its neighbors

**Theorem 2:** Let $G$ be a graph such that
- minimum degree is at least 2
- none of its subgraphs is isomorphic to $K_{1,6}$
- and $G \neq$ 
then $G$ has an $(r, s)$-configuration with $r = \lfloor 5s / 2 \rfloor$.

# Algorithm for Finding an $(r, s)$-configuration

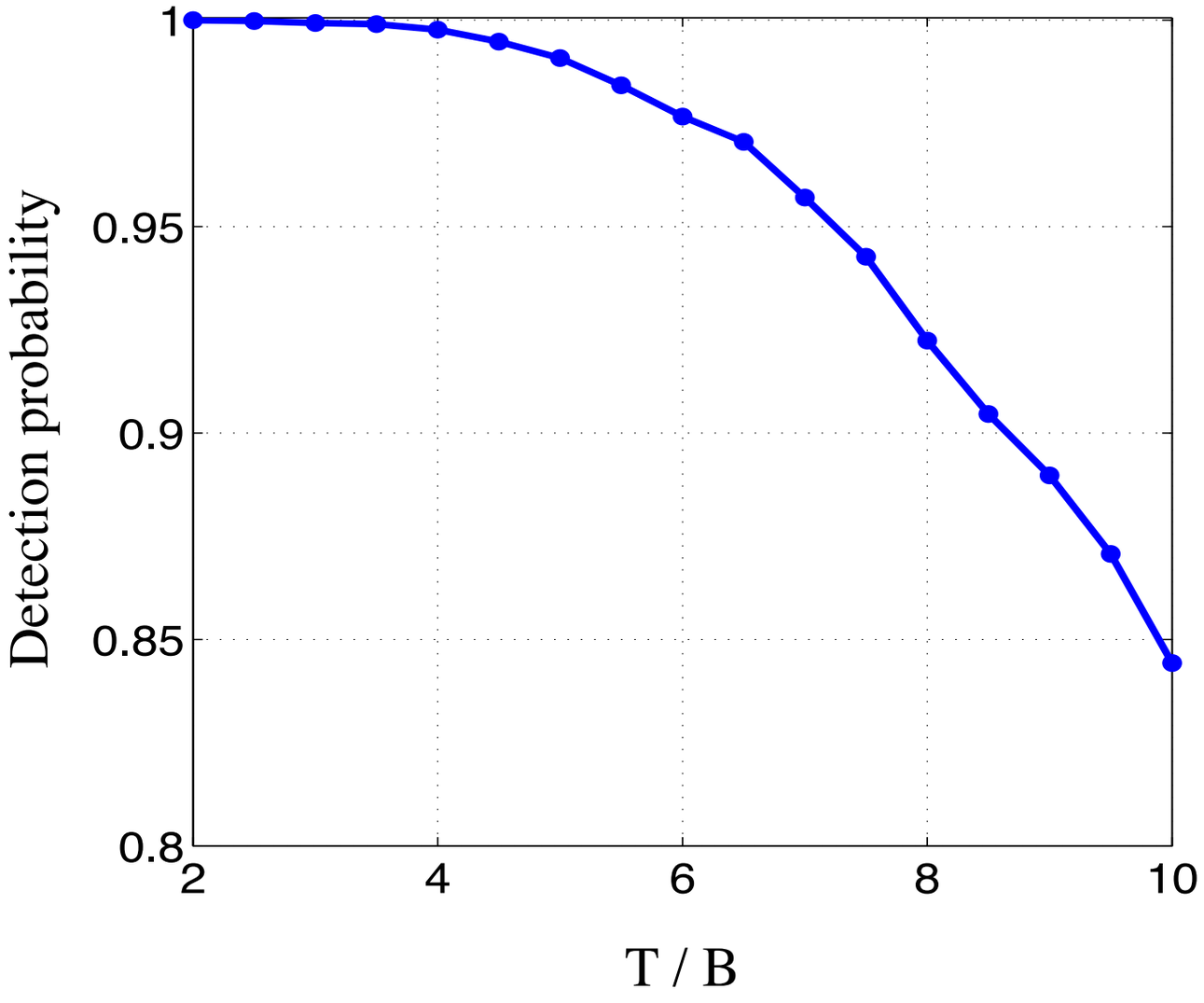
- $A$: set of all $s$ element subsets of the label set $\{1, \ldots, r\}$
- $a_i \in A$: $s$ element subset assigned to node $i$
- $U_i$: number of labels made available by $a_i$ to the neighbors of node $i$ that would not have been available to them otherwise

**Algorithm** Binary Log-Linear Learning

1: **Initialization:** Pick a small $\epsilon \in \mathbb{R}_+$, and a random $a_i \in A$ for every $i \in V$
2: **Repeat**
3: Pick a random node $i \in V$, and a random $a_i' \in A$.
4: Compute $P_\epsilon = \frac{\epsilon^{U_i(a_i', a_{-i})}}{\epsilon^{U_i(a_i', a_{-i})} + \epsilon^{U_i(a_i, a_{-i})}}$.
5: Set $a_i \leftarrow a_i'$ with probability $P_\epsilon$.
6: **End Repeat**

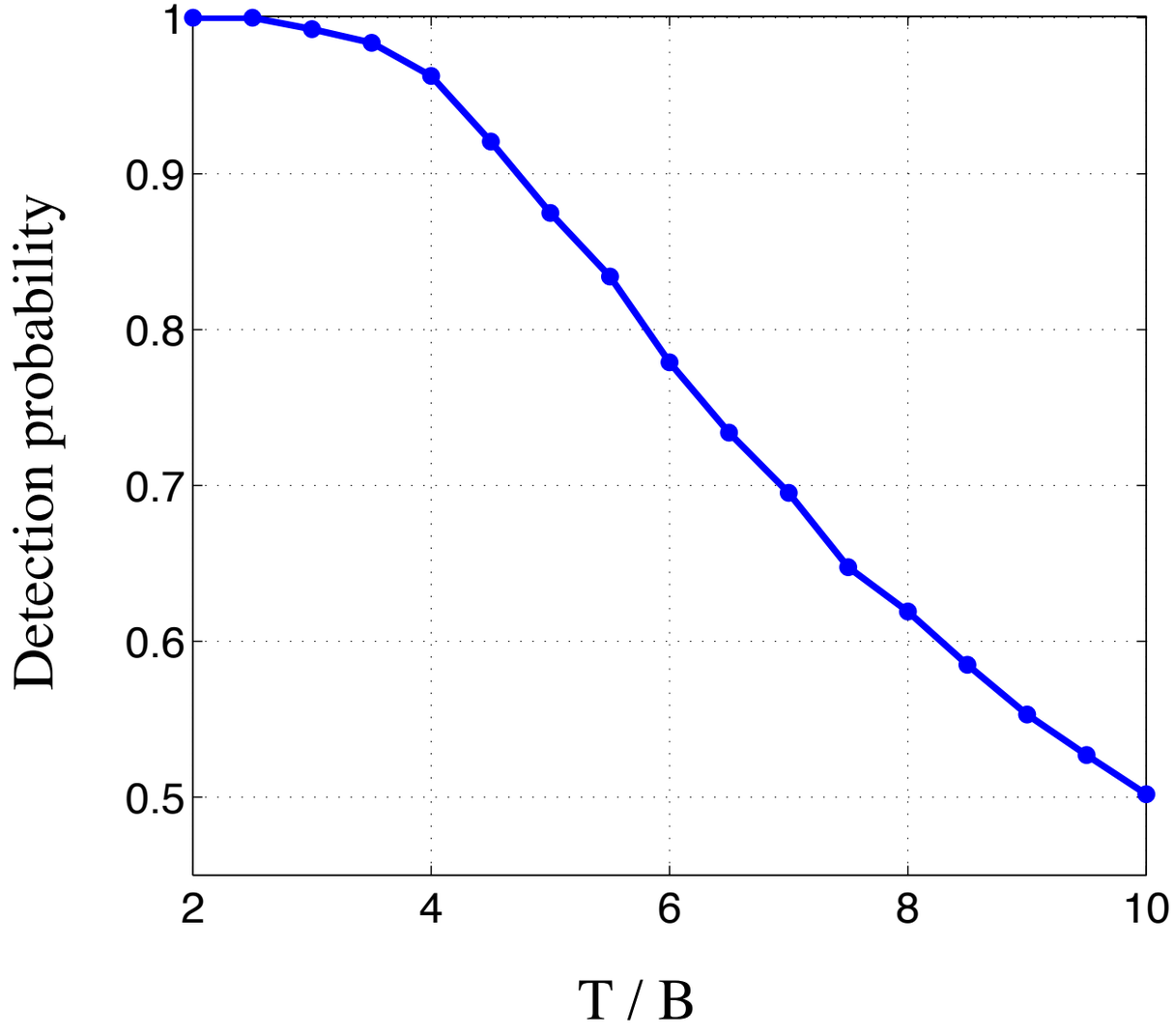
- Support of the limiting distribution converges to the global optimum as the noise parameter approaches zero

# Numerical Results / Geometric Graphs




$S = V$ and $D = 2$

# Numerical Results / Real Water Network




$S = V$ and $D = 2$

# Conclusion and Future Work

- Intrusion detection systems can increase the resilience of cyber-physical systems through early attack detection
- However, running them on resource-bounded devices requires efficient scheduling schemes
- We studied IDS for sensors monitoring water-distribution networks
  - we showed that finding an optimal schedule is NP-hard
  - we proposed heuristic algorithms for worst-case and random attacker
  - we evaluated our algorithms using random graphs and an actual water network
- Future work:
  extend our work towards more general scenarios and physical models of other infrastructure networks

Thank you for your attention!

Questions?