Designing Resilient Cyber-Physical Systems

Aron Laszka
University of Houston

joint work with Waseem Abbas[1], Yevgeniy Vorobeychik[2], and Xenofon Koutsoukos[2]

[1]Information Technology University, Lahore, Pakistan
[2]Vanderbilt University

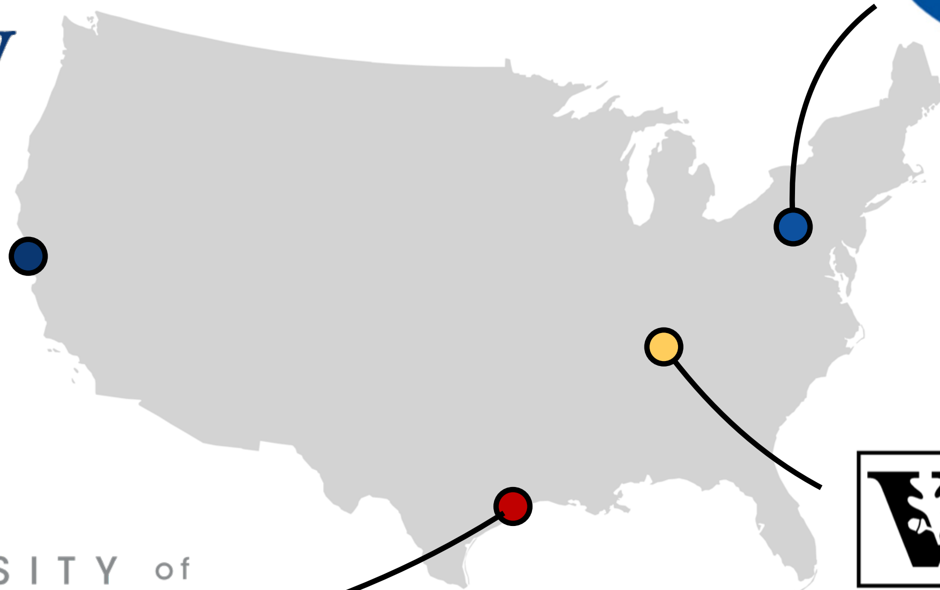
My Background

Berkeley
UNIVERSITY OF CALIFORNIA

Postdoctoral Scholar
2015 - 2016

PennState

Visiting Scholar
2013



VANDERBILT UNIVERSITY

Postdoctoral Research Scholar
2014 – 2015
Research Assistant Professor
2016 – 2017

UNIVERSITY of HOUSTON

Assistant Professor
2017 – present

My Collaborators

Waseem Abbas

Yevgeniy Vorobeychik

Xenofon Koutsoukos

Information Technology University
Lahore, Pakistan

Vanderbilt University
Nashville, TN

Distributed cyber-physical systems, such as smart critical infrastructure, are becoming crucial to everyday life

Cyber-Risks

- Cyber-physical systems are threatened by malicious cyber-attacks, which may have significant physical impact
  - e.g., 2015 and 2016 attacks against Ukrainian power grid
- Defending complex and large-scale CPS, such as smart critical infrastructure, is particularly challenging
  - may contain a number of undiscovered software vulnerabilities due to their sizable codebases
  - large attack surfaces
  - variety of threats
- Example:
  "Dragonfly 2.0" campaign
  - active since 2015
  - targeting energy sector in Europe and North America

Structural Robustness

- Perfect security is virtually impossible in practice

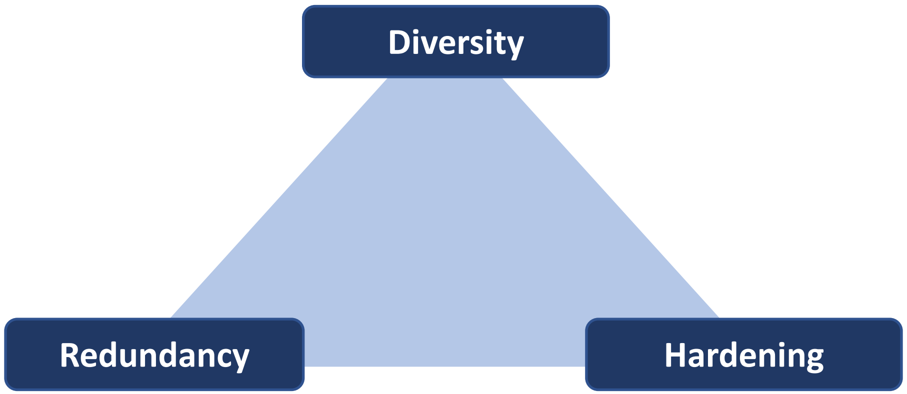
➢ cyber-risks need to be addressed by designing cyber-physical systems to be robust

- **Robustness, resilience, survivability, …**:
  ability of a system to retain its functionality (to some extent) in case of successful cyber-attack

How to improve structural robustness?

Outline

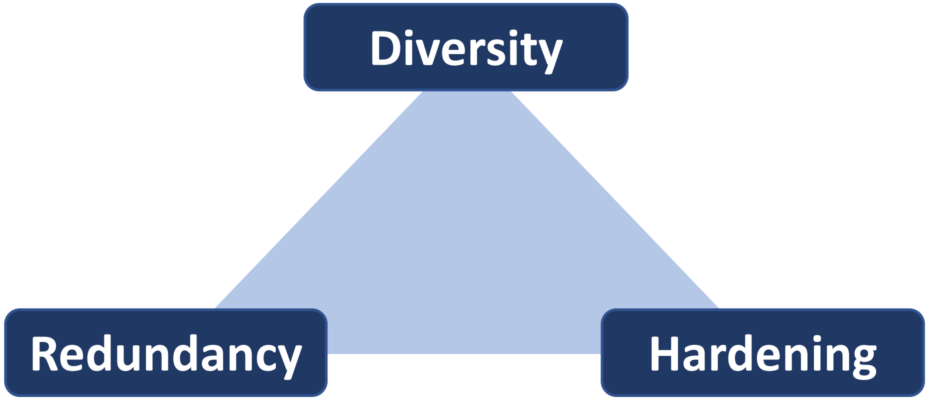
- Structural robustness for distributed CPS
  - redundancy, diversity, and hardening in graphs



- General model and framework for CPS
  - *case studies*: cyber-physical attacks against smart water-distribution and cyber-attacks against transportation
- Conclusion and future work

Improving Structural Robustness

- Canonical approaches:



  - <u>Redundancy</u>: deploying additional, redundant components in a system, so even if some components are compromised or impaired, the system may retain correct functionality
  - <u>Diversity</u>: implementing the components of a system using a diverse set of component types, so that vulnerabilities that are present in only a single type have limited impact
  - <u>Hardening</u>: reinforcing individual components or component types (e.g., tamper-resistant hardware and firewalls)

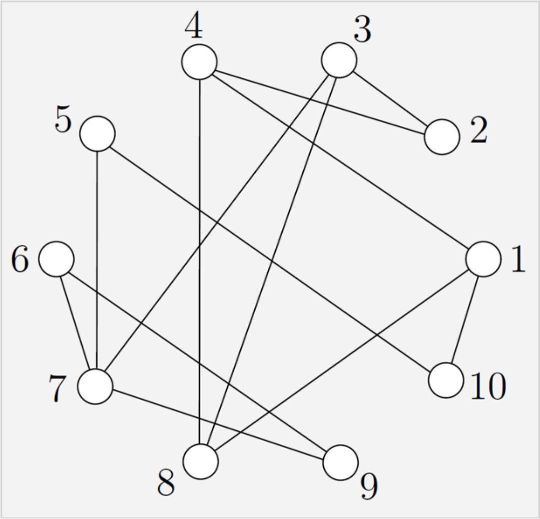
How to combine redundancy, diversity, and hardening?

Example: Improving Network Availability

- **Pairwise connectivity**: fraction of node pairs that are connected with each other through a path
  - we use it to measure network availability
- **Simple attack model**: adversary removes N nodes to minimize the pairwise connectivity of the residual network

- Example:

pairwise connectivity = 1

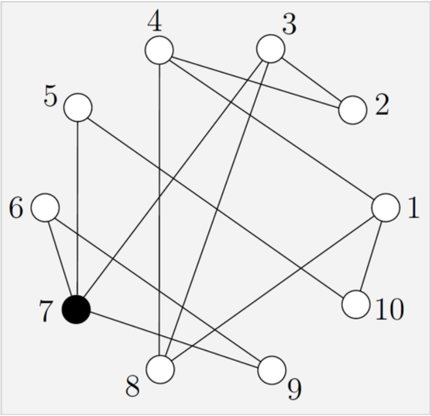


- worst-case N = 2 attack removes nodes {1, 7}
- pairwise connectivity after attack = **0.286**

Hardening and Diversity

- **Hardening**: protect a subset of nodes from attacks

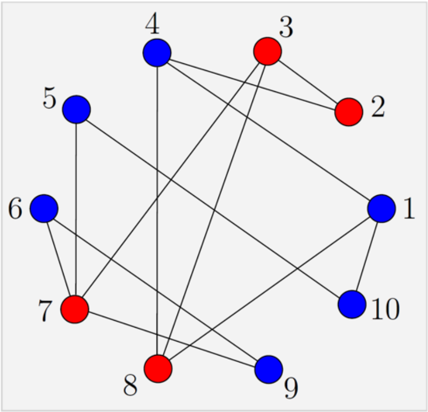
node 7 is hardened



- worst-case N = 2 attack removes nodes {3, 10}
- pairwise connectivity after attack = **0.429** (> 0.286)

- **Diversity**: each node has a type, and the adversary can attack nodes of only one type

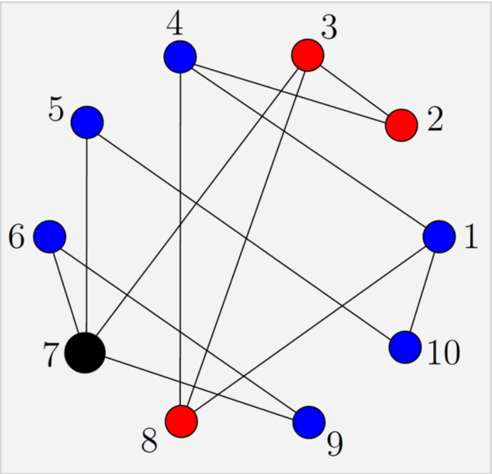
two types,
red and **blue**



- worst-case N = 2 attack removes nodes {2, 7}
- pairwise connectivity after attack = **0.571** (> 0.286)

Combining Hardening and Diversity

- two types, **red** and **blue**
- node 7 is hardened

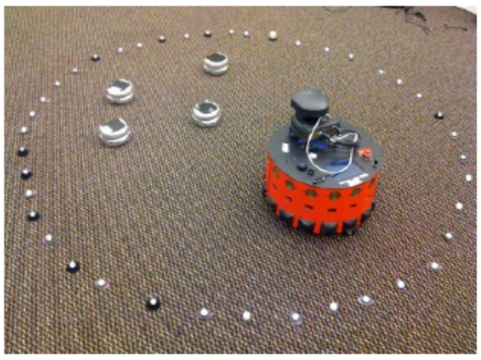


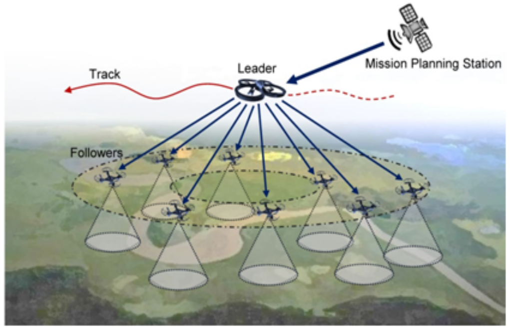
- worst-case N = 2 attack removes nodes {1, 5}
- pairwise connectivity after attack = **0.75** (> 0.571)

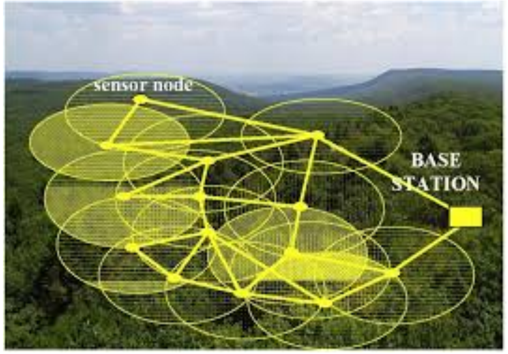
What about integrity?

Networked Systems

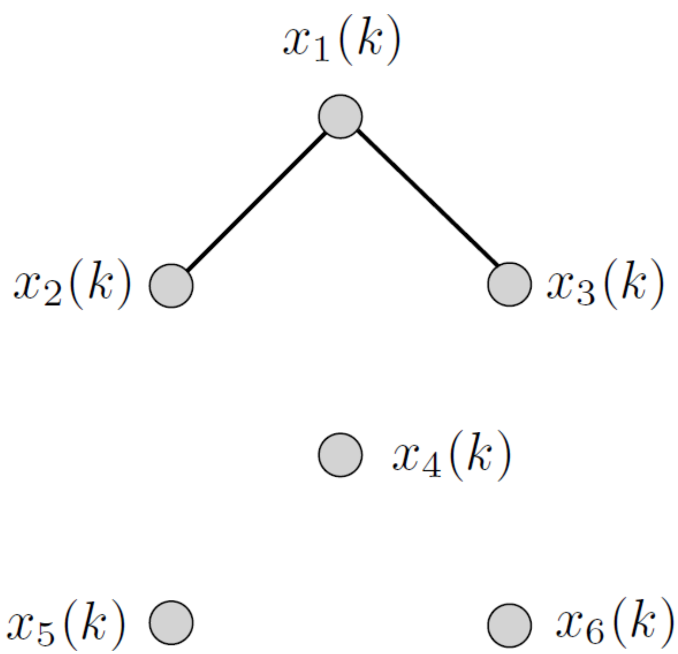
- In many networked control systems, a **global objective** needs to be achieved through **local interactions**
- The individual components have **limited sensing**, **computational**, and **communication capabilities**





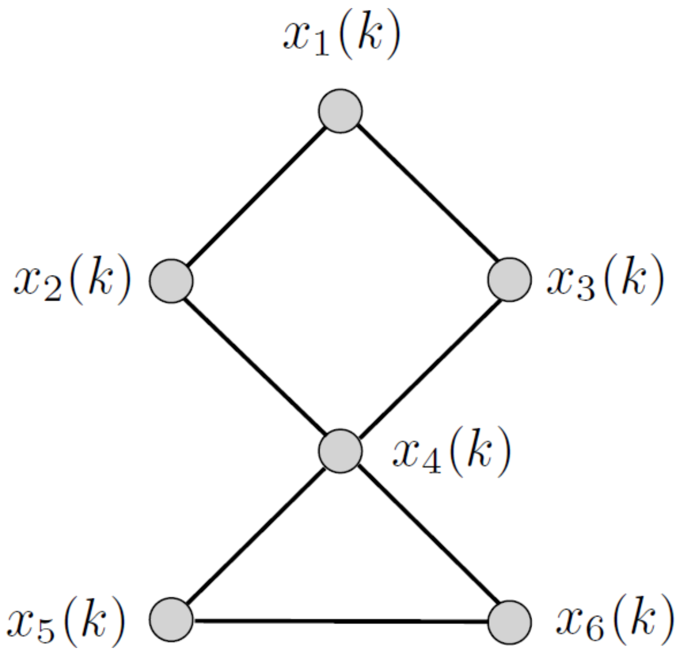
Global Objective through Local Interactions



$$x_1(k+1) \;=\; f(x_1, x_2, x_3)$$

$x_i(k)$: state of node i at time step k

Global Objective through Local Interactions



$$\left.\begin{aligned} x_1(k+1) &= f(x_1, x_2, x_3) \\ x_2(k+1) &= f(x_1, x_2, x_4) \\ x_3(k+1) &= f(x_1, x_3, x_4) \end{aligned}\right\} \text{Local interactions}$$

$$\vdots$$

Global objective is a function of

$$\mathbf{X} = (x_1, x_1, \cdots, x_7)$$

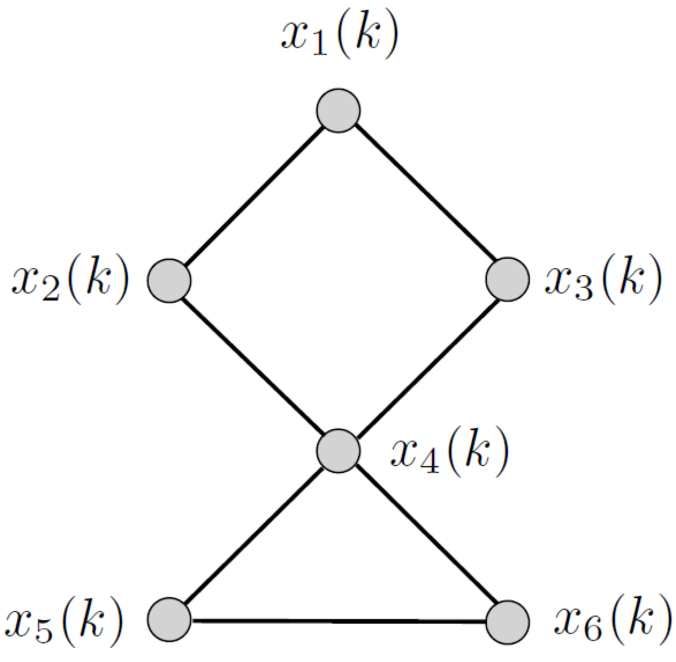
$x_i(k)$: state of node i at time step k

Consensus Problem

- Canonical problem formulation: **Consensus Problem**

All nodes need to eventually converge to a common state:

$$\lim_{k \to \infty} x_i(k) = x, \forall i$$



$$x_i(k+1) = \sum_{j \in N_i(k)} w_{ij}(k) x_j(k)$$

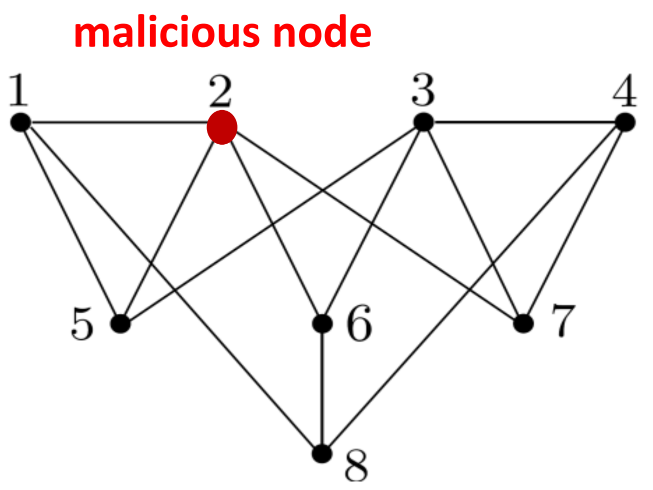
Linear Consensus Protocol (LCP)

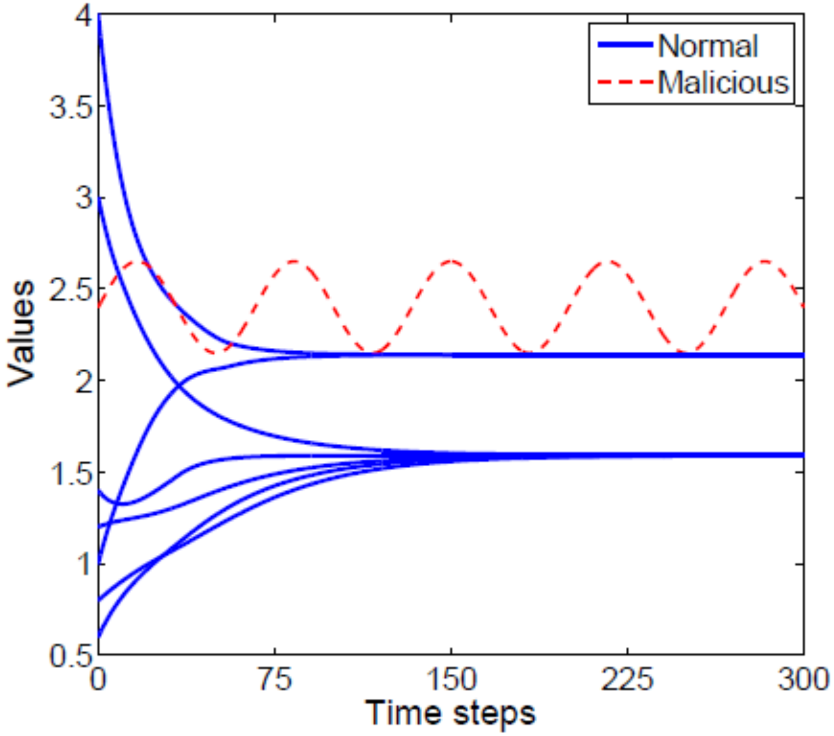
- consensus is achieved if all nodes implement LCP, and the underlying graph is connected

Resilient Consensus Problem

- Malicious nodes: their goal is to prevent the network from reaching consensus (e.g., compromised by an adversary)

- Example





Resilient Consensus Problem (contd.)

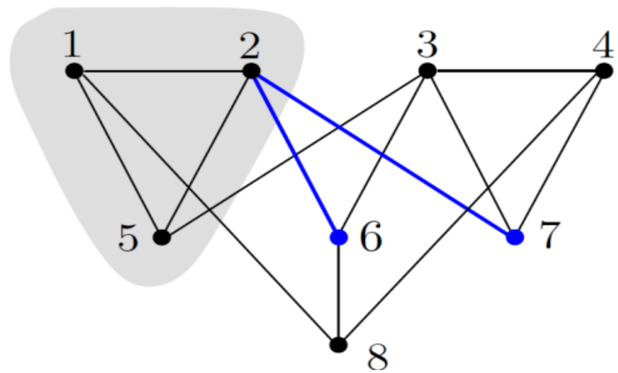
- Models
  - ***F-total malicious model***:
    if $S \subseteq V$ is the set of malicious nodes, then $|S| \leq F$
  - ***F-local malicious model***:
    if $S \subseteq V$ is the set of malicious nodes, then $|N(i) \cap S| \leq F$,
    for every $i \in V \setminus S$

Goal:
characterize networks in which nodes can reach consensus under the F-total or F-local malicious models

- Previous work: r-robustness and (r,s)-robustness

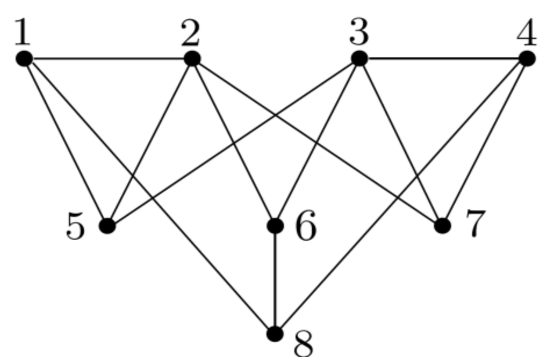
r-Robustness

- **r-reachable subset**:
  a subset of nodes S is r-reachable if there exists at least one node in S that has at least r neighbors outside of S



subset S = {1, 2, 5}
is 2-reachable

- **r-robust graph**:
  a graph is r-robust if for any pair of non-empty and disjoint subsets of nodes, at least one of them is r-reachable

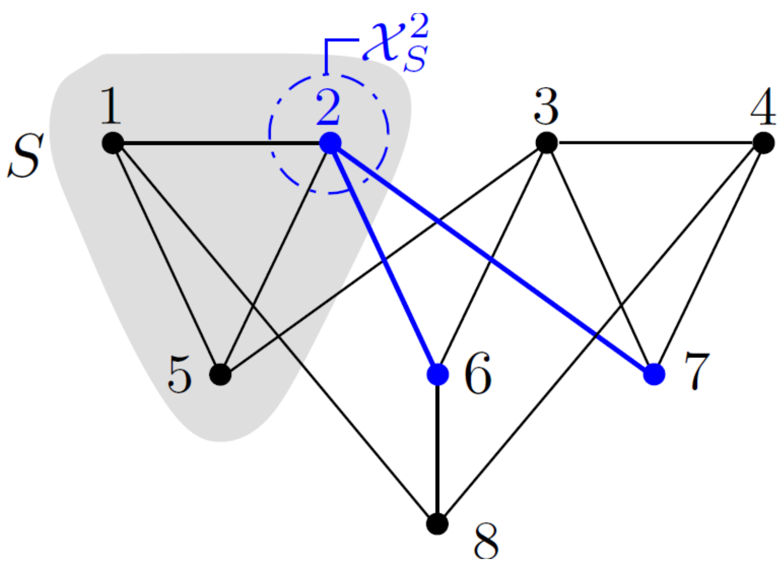


2-robust graph

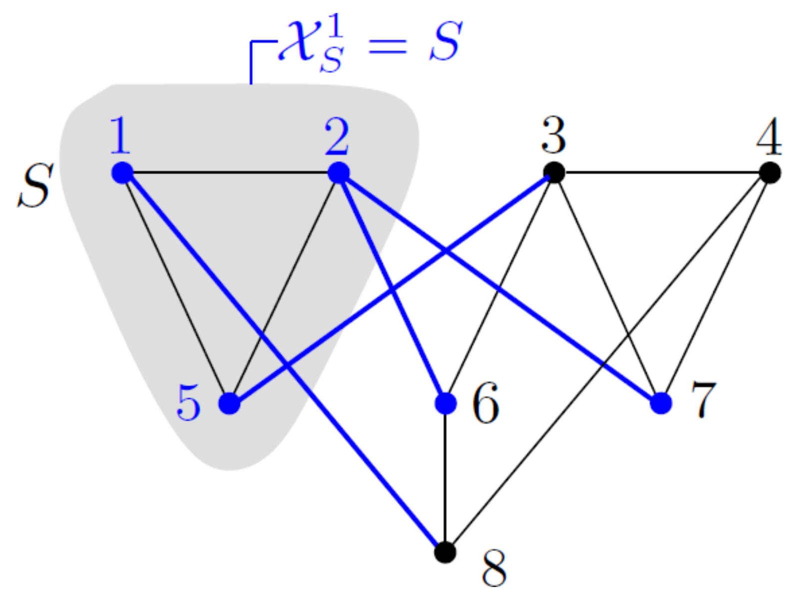
(r,s)-Robustness

- Let S be a set of nodes, then $\mathcal{X}_S^r$ is the subset of nodes in S that each have at least r neighbors outside of S

$$\mathcal{X}_S^r = \{v \in S : |N(v) \cap (V \setminus S)| \geq r\}$$



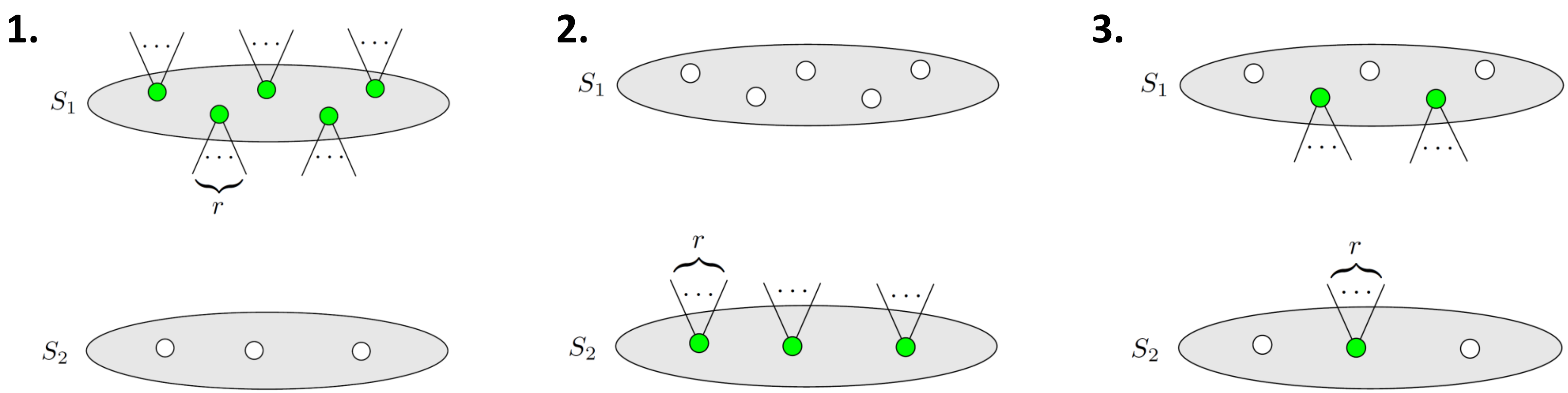
$\mathcal{X}_S^2 = \{2\}$



$\mathcal{X}_S^1 = \{1, 2, 5\} = S$

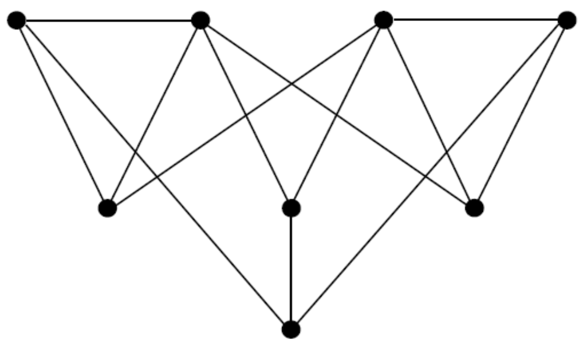
(r,s)-Robustness (contd.)

- **(r,s)-robust graph**:
  A graph is (r,s)-robust if for every pair of non-empty, disjoint subsets S_1 and S_2 of V, at least one of the following holds:
  1. $|\mathcal{X}_{S_1}^r| = |S_1|$
  2. $|\mathcal{X}_{S_2}^r| = |S_2|$
  3. $|\mathcal{X}_{S_1}^r| + |\mathcal{X}_{S_2}^r| \geq s$
- **r-robust = (r, 1)-robust**



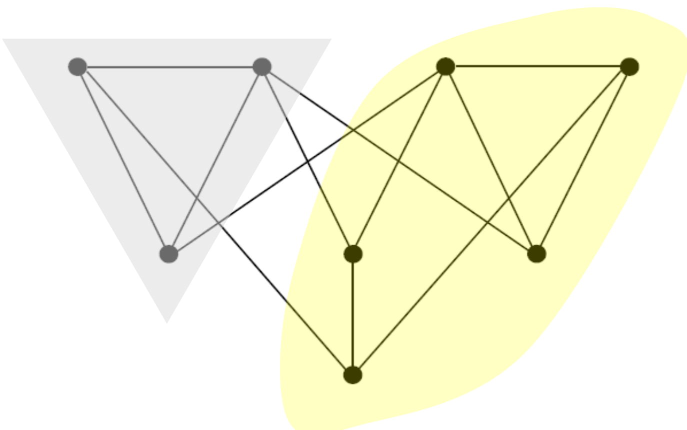
number of green nodes ≥ s

Examples of (r,s)-Robust Graphs

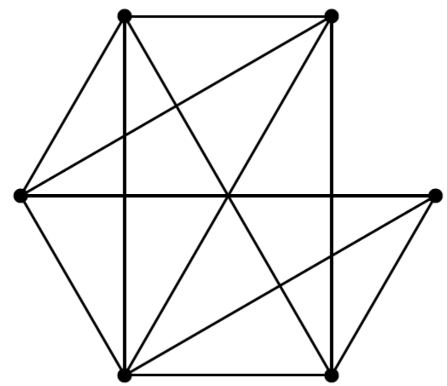


(2,1)-robust
(hence, 2-robust)

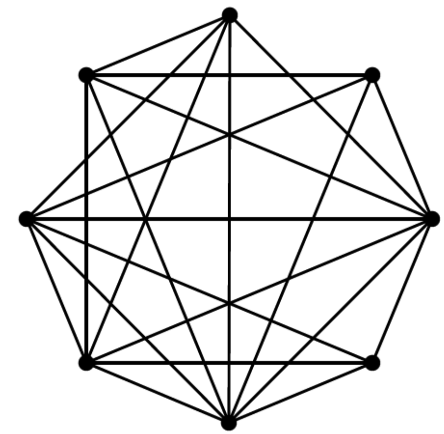
Examples of (r,s)-Robust Graphs



Not (2,2)-robust



(2,2)-robust



(3,3)-robust

(r,s)-Robustness and Resilient Consensus

Theorem (LeBlanc et al. 2013):
Let G(V, E) be a time-invariant network in which each normal node implements the Weighted-Mean- Subsequence-Reduced (WMSR) algorithm. Then,

1. under the ***F-total malicious model***, consensus is achieved asymptotically if and only if G is **(F + 1, F + 1)-robust**
2. under the ***F-local malicious model***, to achieve asymptotic consensus, it is necessary that G is (F + 1)-robust, and is sufficient that G is **(2F + 1)-robust.**

- WMSR idea:
  omit F lowest and F highest values from state update

Hardening: Trusted Nodes

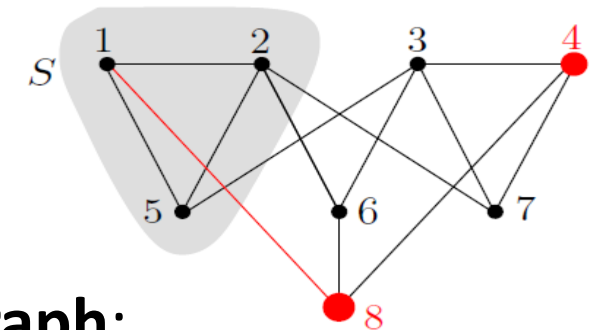
- Unfortunately, r-robustness is a very strong property
  - some graphs have very large connectivity but low robustness
- In practice, increasing connectivity through deploying a large number of new nodes and links may be **impossible** or **prohibitively expensive**
- **Hardening**: instead of increasing connectivity, we make a small set of nodes **trusted**
  - trusted nodes are protected from adversaries
  - for example, tamper-resistant hardware, complex firewalls, physical protection

Goal:
characterize networks in which nodes can reach consensus with the help of trusted nodes

r-Robustness with Trusted Nodes

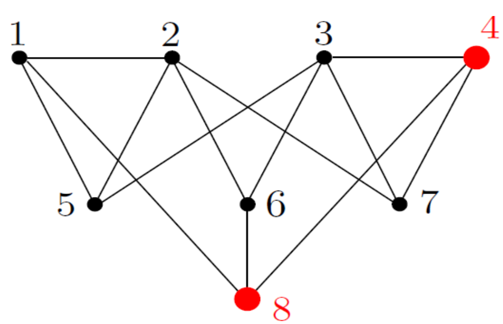
- **r-reachable subset with trusted nodes T**:
  a subset of nodes S is r-reachable with trusted nodes T if there exists at least one node in S that has at least r neighbors outside of S or one trusted neighbor outside of S

T = {4, 8}



subset S = {1, 2, 5} is not 3-reachable, but it is 3-reachable with trusted nodes

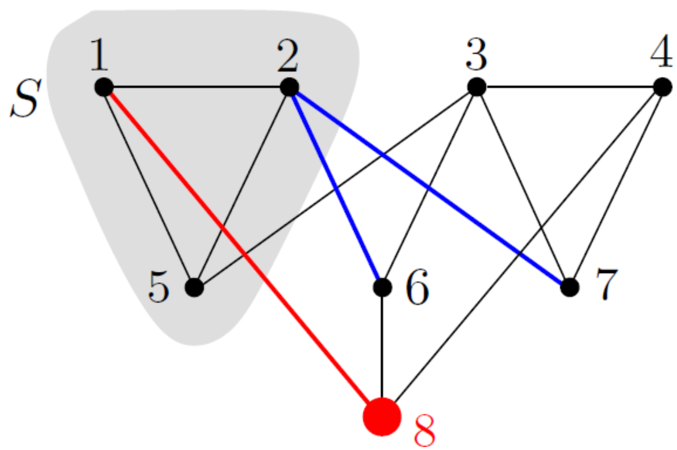
- **r-robust graph**:
  graph is r-robust with trusted nodes if for any two non-empty and disjoint subsets of nodes, at least one of them is r-reachable with trusted nodes



3-robust graph with trusted nodes

(r,s)-Robustness with Trusted Nodes

- Let S be a subset of nodes, then $\mathcal{Z}_S^r$ is a subset of S such that each node in $\mathcal{Z}_S^r$ has at least r neighbors outside of S or one trusted neighbor outside of S



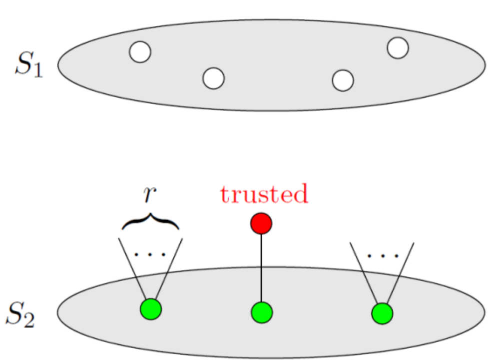
T = {8}

- for S = {1, 2, 5}, we have $\mathcal{Z}_S^2$ = {1, 2}
  since node 2 has two neighbors outside of S, and node 1 has a trusted neighbor outside of S

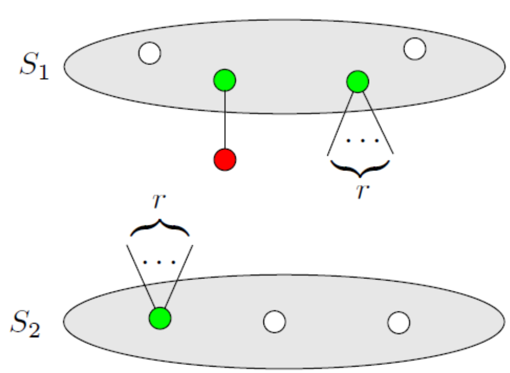
(r,s)-Robustness with Trusted Nodes (contd.)

- **(r,s)-robust graph with trusted nodes**:
  A graph is (r,s)-robust with trusted nodes T if for every pair of non-empty, disjoint subsets S_1 and S_2 of V, at least one of the following holds:
  1. $|\mathcal{Z}_{S_1}^r| = |S_1|$
  2. $|\mathcal{Z}_{S_2}^r| = |S_2|$
  3. $|\mathcal{Z}_{S_1}^r| + |\mathcal{Z}_{S_2}^r| \geq s$
  4. $(\mathcal{Z}_{S_1}^r \cup \mathcal{Z}_{S_2}^r) \cap T \neq \emptyset$

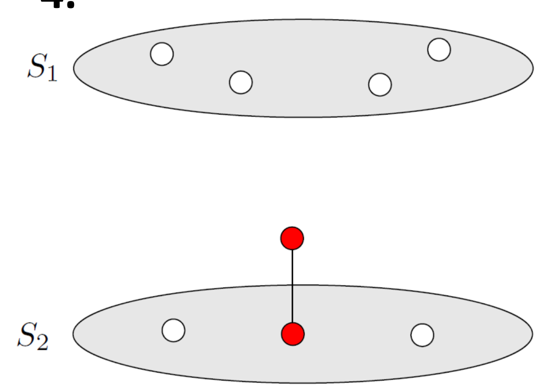
2.



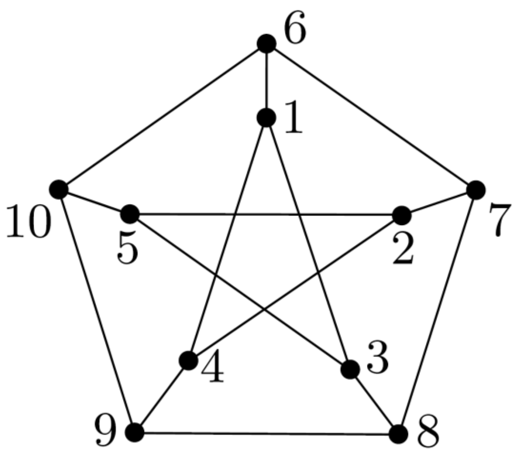
3.



4.

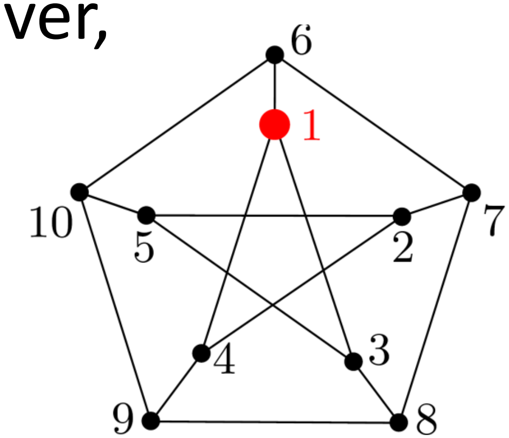


Example (r,s)-Robust Graphs with Trusted Nodes

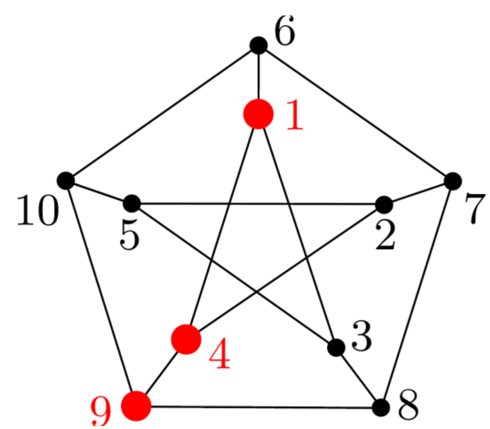


- Peterson graph is not 2-robust
- For instance, consider $S_1 = \{1, 2, 3, 4, 5\}$; $S_2 = \{6, 7, 8, 9, 10\}$
- Neither of these subsets contains a node that has two neighbors outside of the subset

- However,

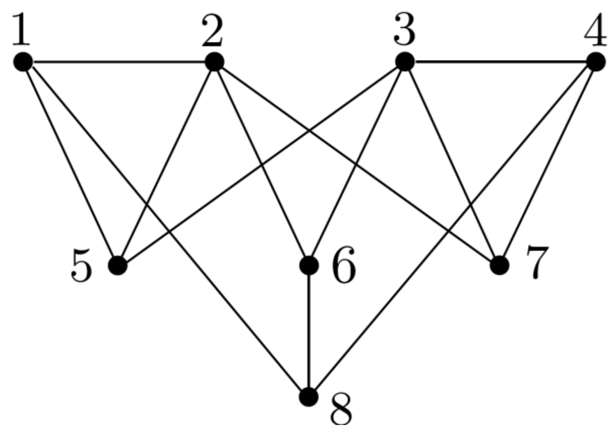


graph is **2-robust** with any single node as trusted node



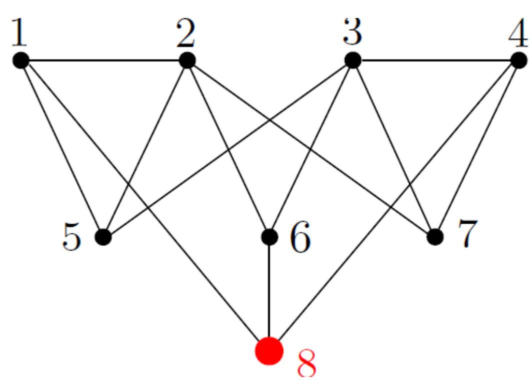
graph is **3-robust** with trusted nodes {1, 4, 9}

Example (r,s)-Robust Graphs with Trusted Nodes

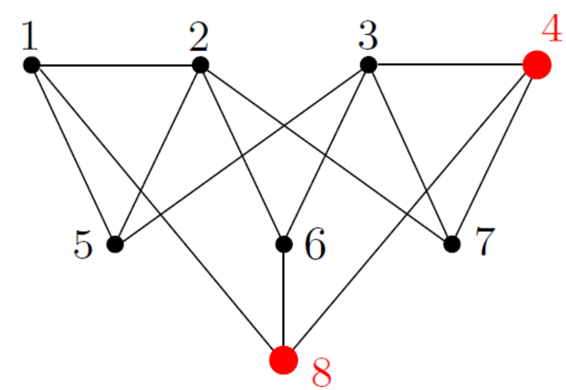


- Graph is 2-robust, but not (2,2)-robust
- For instance, consider
  S_1 = {1, 2, 3, 5};
  S_2 = {3, 4, 6, 7, 8}

- However,



graph is **(2,2)-robust** with a single trusted node T = {8}



graph is **3-robust** with trusted nodes T = {4, 8}

Robustness with Trusted Nodes and Resilient Consensus

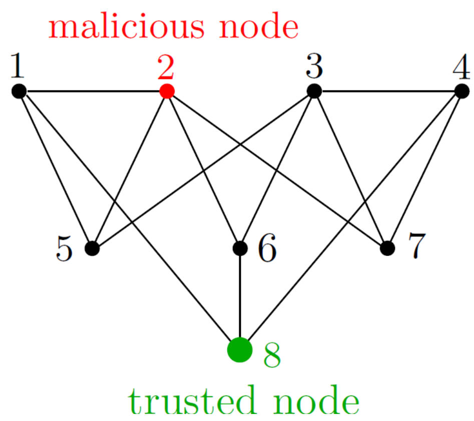
- Results that relate (r,s)-robustness to the resilience of consensus can be generalized using the notion of **(r,s)-robustness with trusted nodes**

Theorem:
Let G(V, E) be a time-invariant network with trusted nodes T in which each normal node implements the RCA-T algorithm. Then,

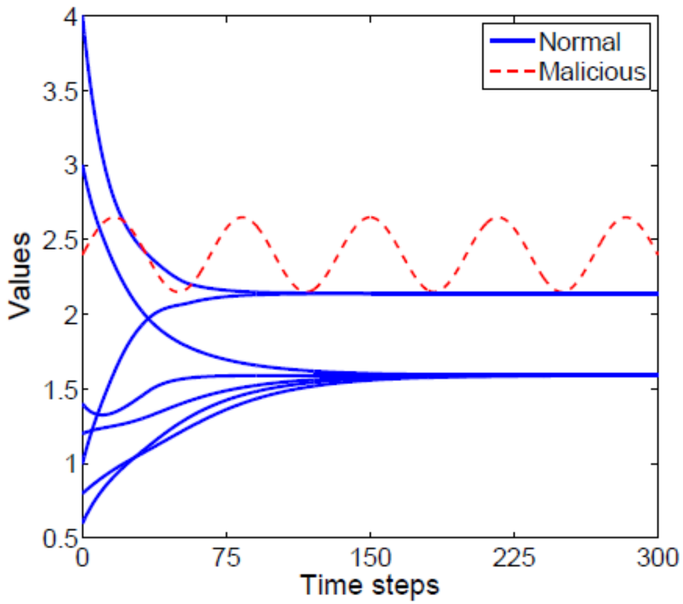
1. under the *F-total malicious model*, consensus is achieved asymptotically if and only if G is **(F + 1, F + 1)-robust with T**.
2. under the *F-local malicious model*, to achieve asymptotic consensus, it is necessary that G is **(F + 1)-robust with T**, and is sufficient that G is **(2F + 1)-robust with T.**

- Resilient Consensus Algorithm with Trusted nodes (RCA-T): always accept values for state update from trusted nodes

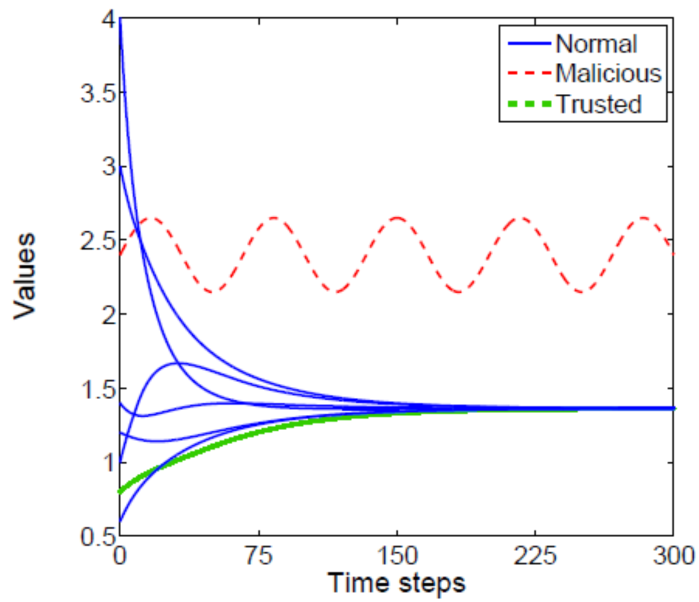
Illustration for F-Total Model



- G is (2,2)-robust with T = {8}
- There is one malicious node.

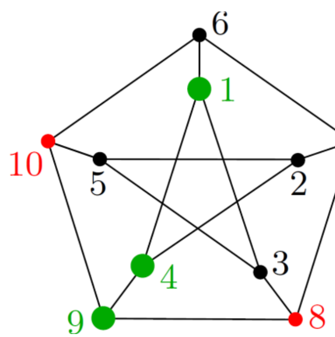


WMSR – algorithm:
consensus cannot be achieved

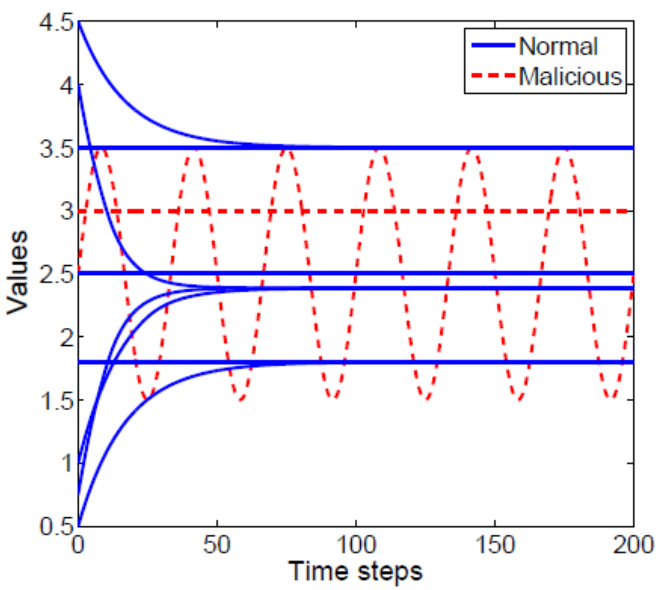


RCA-T – algorithm:
consensus is achieved with trusted node

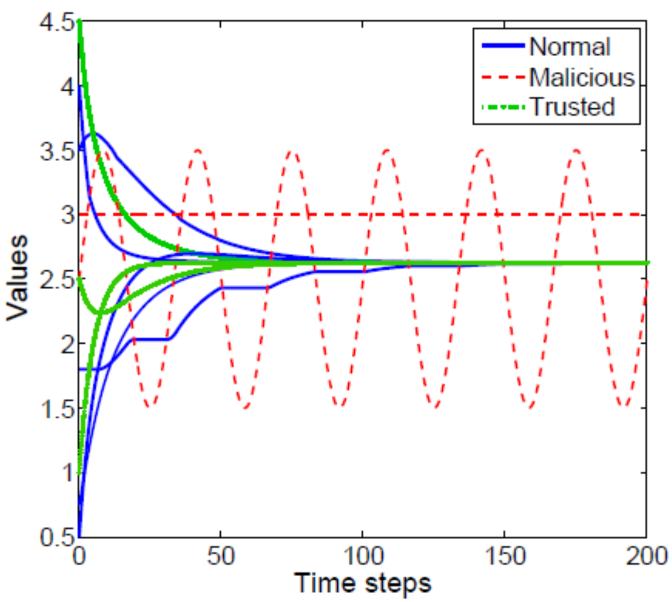
Illustration for F-Local Model



- G is 3-robust with T = {1, 4, 9}
- There are two malicious nodes which are {8, 10}



WMSR – algorithm:
consensus cannot be achieved



RCA-T – algorithm:
consensus is achieved with trusted nodes

Building Robust Graphs

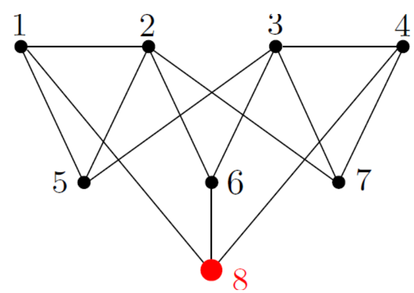
Adding Nodes to Robust Graphs

Theorem:

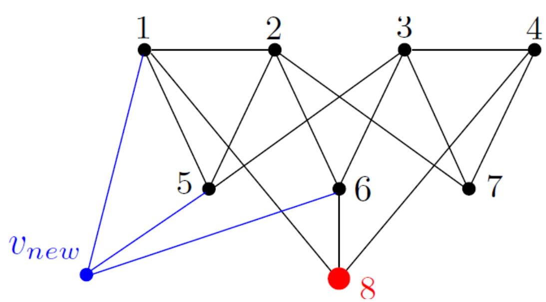
Let G be (r,s)-robust with trusted nodes, then adding a new node v_{new} to G preserves the robustness property of the graph if

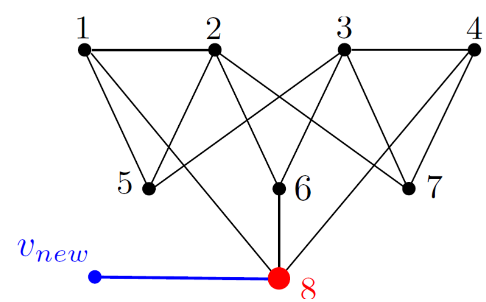
1. v_{new} is adjacent to at least (r+s-1) non-trusted nodes, or
2. v_{new} is adjacent to at least one trusted node.

Example:



(2,2)-robust graph with the red trusted node





- v_{new} is connected to 3 non-trusted nodes
- New graph is still (2,2)-robust

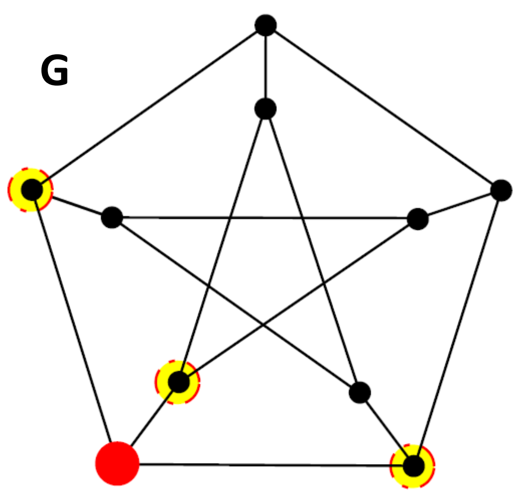
- v_{new} is connected to a single trusted node
- New graph is still (2,2)-robust

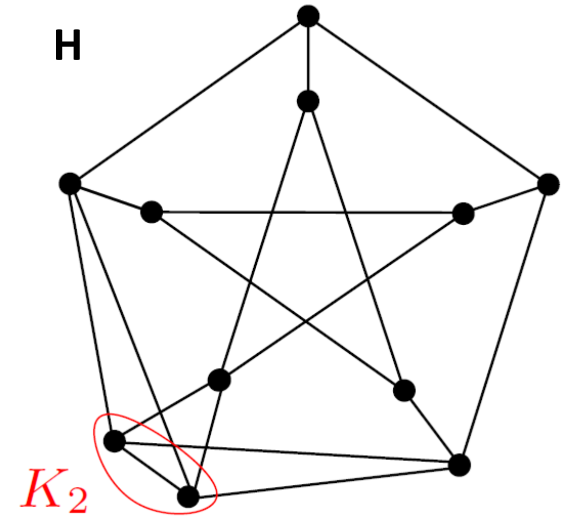
Replacing Trusted Node with Clique

Theorem:

Let G be an r-robust graph with trusted nodes T. Let t ∈ T, and H be a graph obtained by replacing t with a clique of size r, denoted by K_r, such that each neighbor of t in G is adjacent to each node in K_r, then H is also r-robust.

Example:





- A **2-robust** graph with a red trusted node
- Neighbors of trusted node are highlighted

- A trusted node is replaced by K_2
- **H** is still **2-robust**

Replacing Trusted Node with Robust Graph

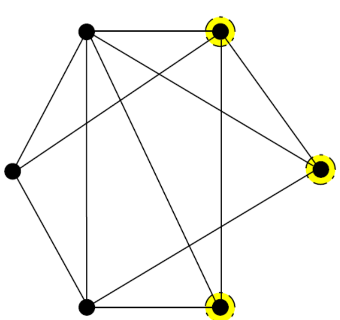
Theorem:

Let G be an r-robust graph with trusted nodes T, G’ be another r-robust graph, and η be a non-reachable subset of nodes in G’.

Let t ∊ T, and H be a graph obtained from G by replacing t with G’ such that each neighbor of t in G is adjacent to each node in the subset η of G’, then H is also r-robust.

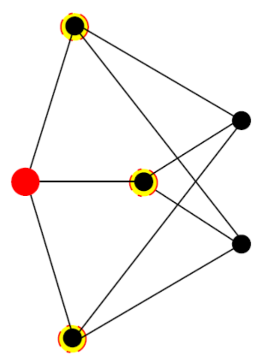
Example:

G’



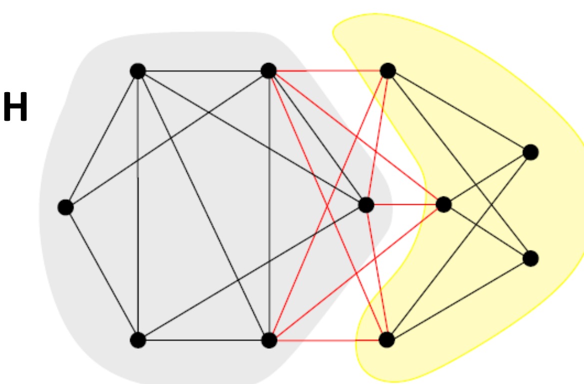
- G’ is **3-robust**
- Nodes in subset η are highlighted

G



- G is **3-robust** with red trusted node
- Neighbors of trusted node are highlighted

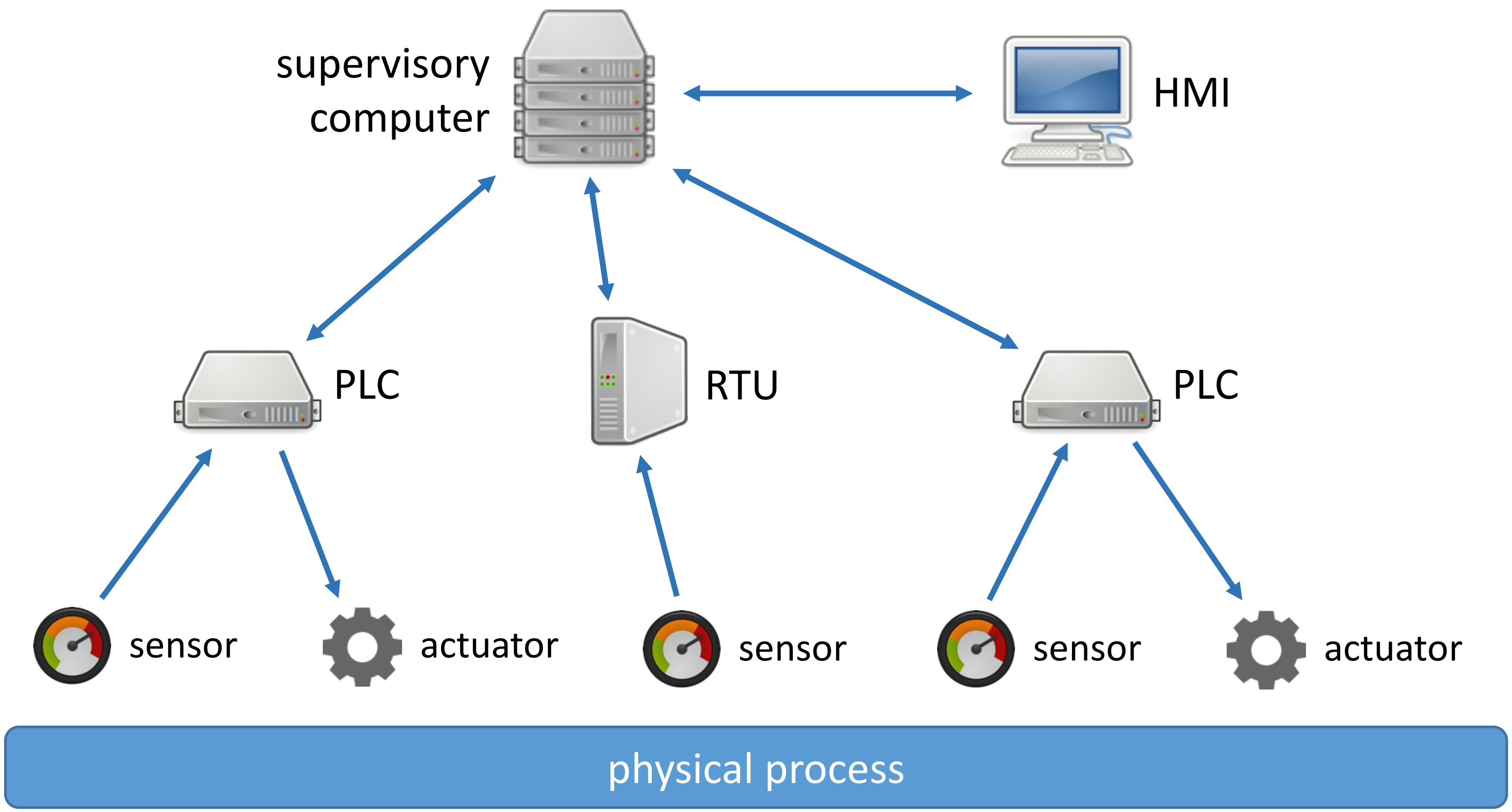
H



- H is also is **3-robust**
- New edges added are shown in red

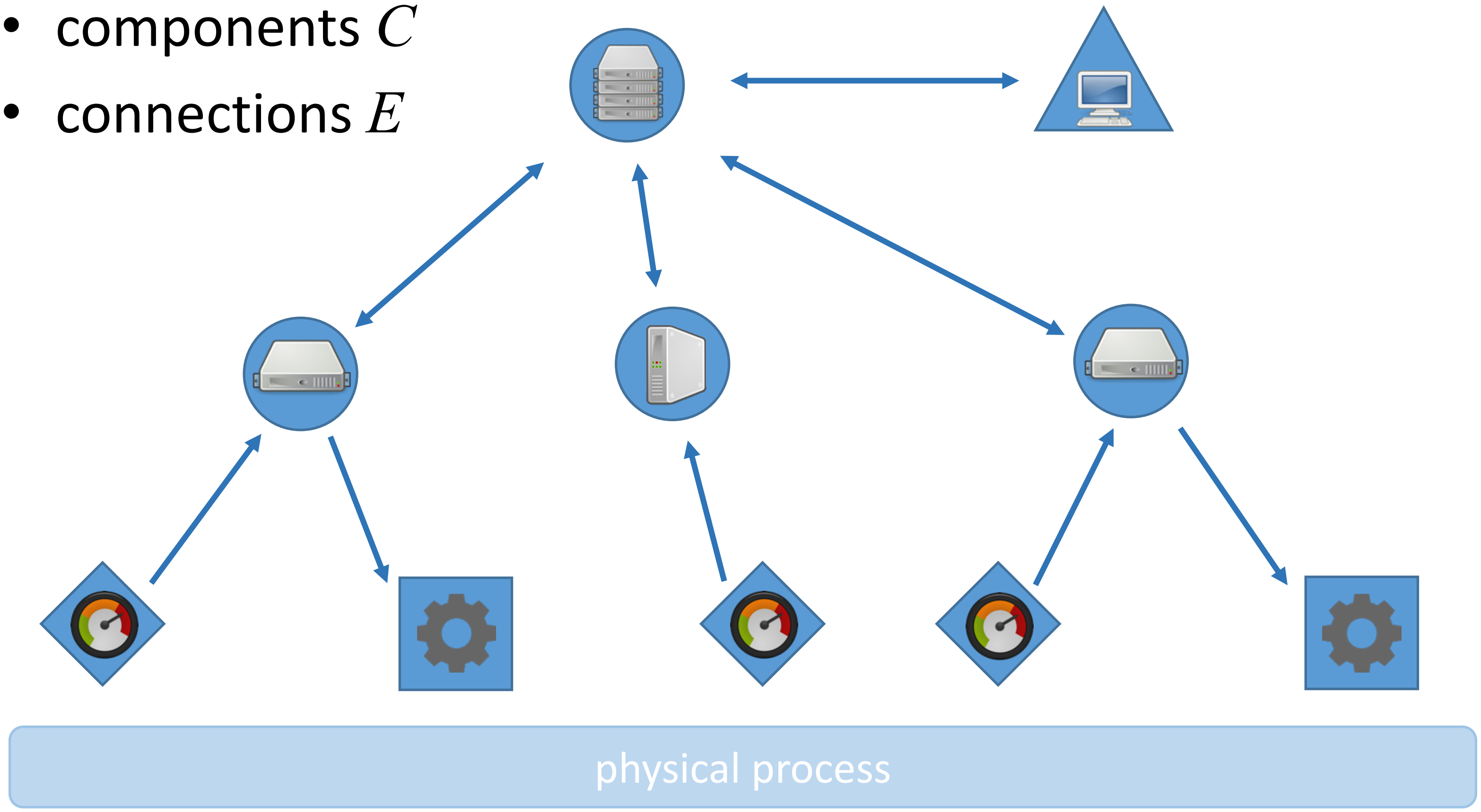
General Framework for Cyber-Physical Systems

Example Cyber-Physical System



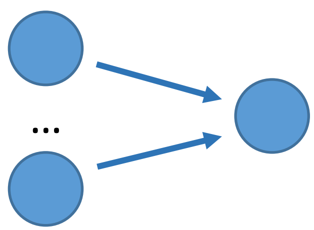
Graph-Theoretic Model

- Graph $G = (C, E)$
  - components C
  - connections E

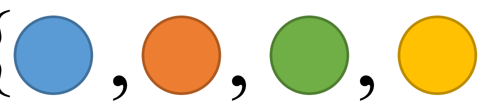


Components

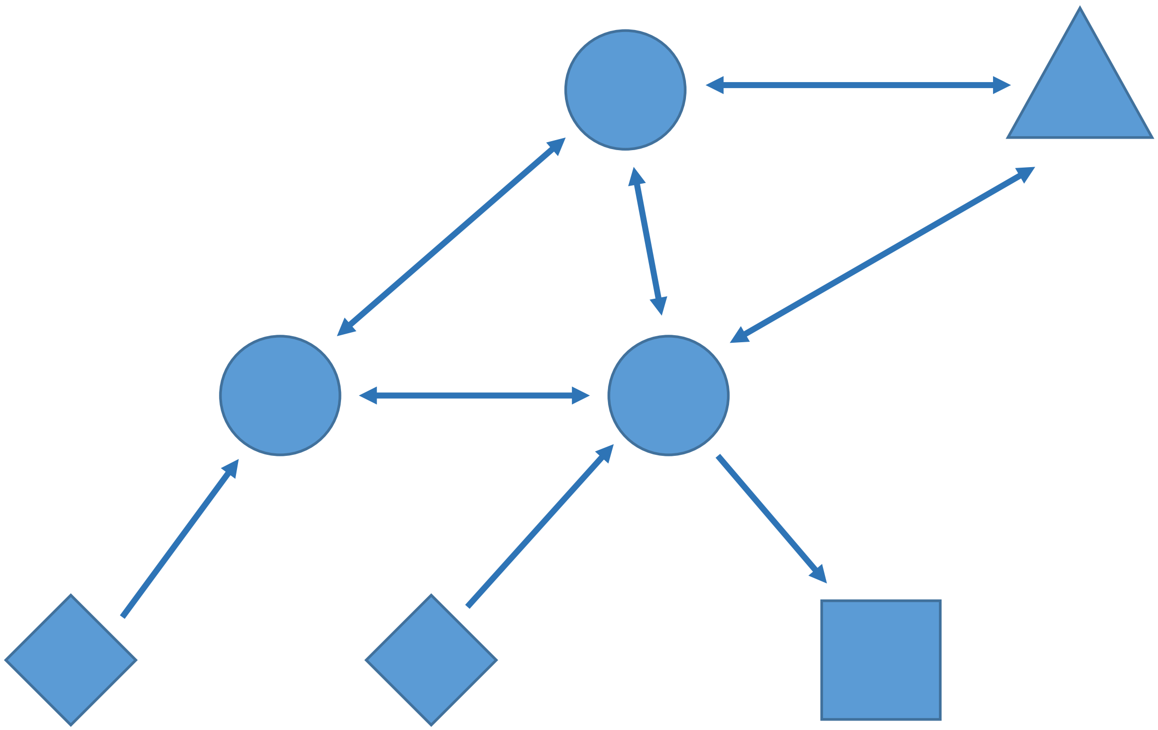
- Properties of a component $c \in C$
  - type t_c
    - computational
    - sensor
    - actuator
    - interface
  - set of input connections E_c
    - example:



  - deployed implementation r_c
    - chosen from a set of available implementations I
    - example set:

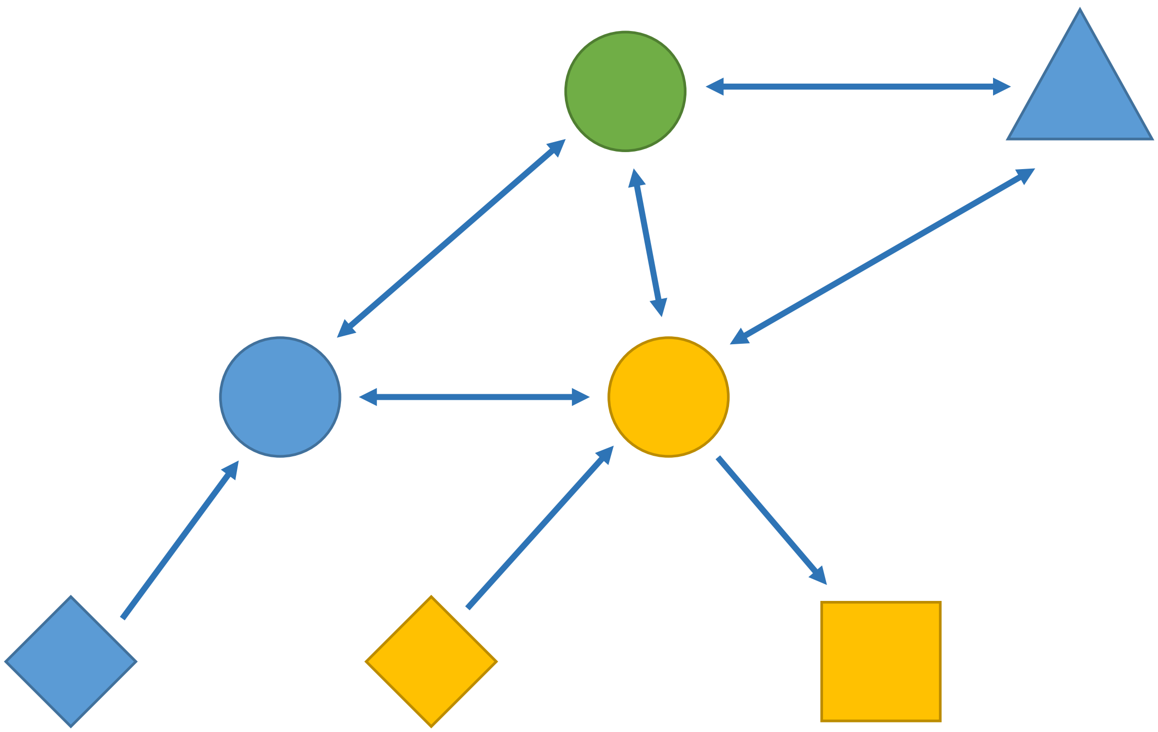
$I = \{$  $\}$

How to improve the resilience of a CPS?



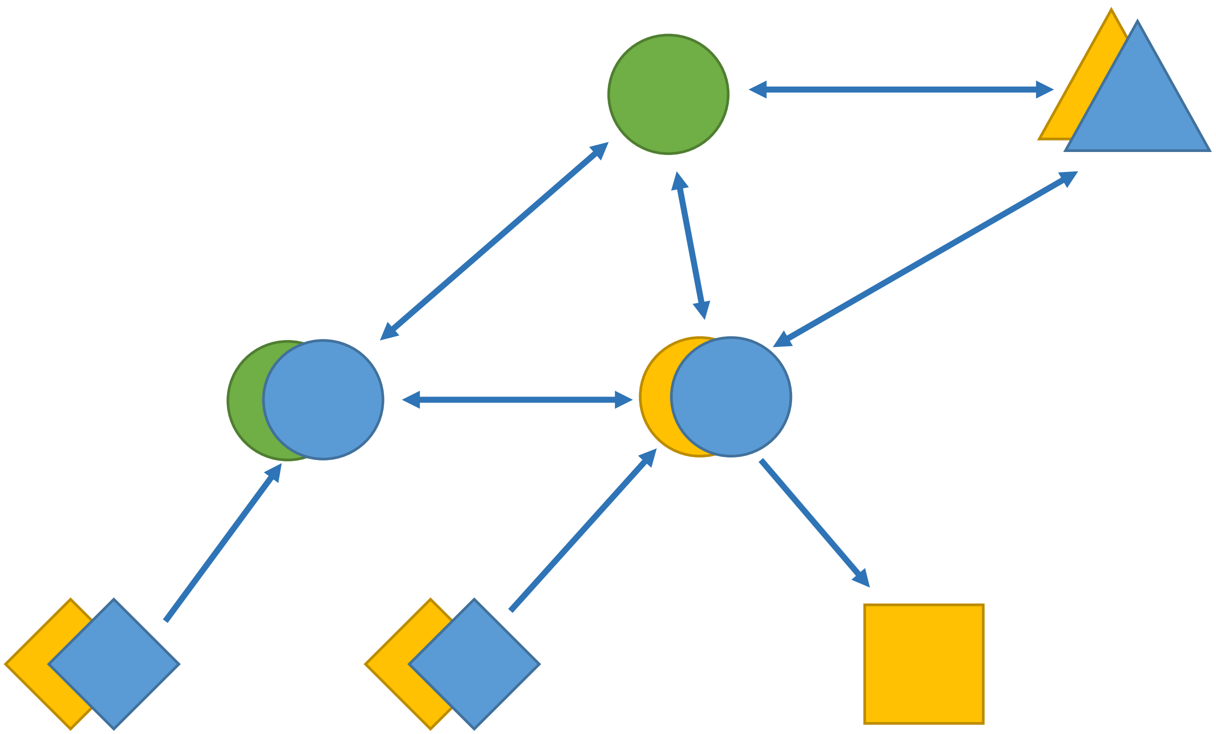
Diversity

- use a variety of implementations
- each implementation $i \in I$ has a usage cost D_i

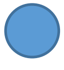


Redundancy

- deploy additional instances of some components (based on different implementations)
- each implementation $i \in I$ has a deployment cost R_i



Hardening

- Harden some implementations (e.g., source code reviews, firewalls, penetration testing)
- Each implementation has a set of available hardening levels L_i
  - each level $l \in L_i$ has a cost H_l and an estimate of being secure S_l
  - example levels:
    { (DEFAULT: \$100 000, 0.9),
    (SECURE: \$500 000, 0.95),
    (VERY SECURE: \$1 000 000, 0.99) }
- Example selection:
  -  → SECURE
  -  → DEFAULT
  -  → VERY SECURE

Resilience Maximization Problem

- Given redundancy, diversity, and hardening expenditures $\boldsymbol{R}$, $\boldsymbol{D}$, $\boldsymbol{H}$, the optimal deployment is

$$\min_{\boldsymbol{r},\, \boldsymbol{l}} \mathrm{Risk}(\boldsymbol{r}, \boldsymbol{l})$$

subject to $\sum_{c\in C}\sum_{i\in r_c} R_i \leq \boldsymbol{R},\ \sum_{i\in \bigcup_c r_c} D_i \leq \boldsymbol{D},\ \sum_{i\in I} H_{l_i} \leq \boldsymbol{H}$

- Computationally challenging (NP-hard), but we have efficient heuristics that work well in practice
- General problem: given budget $\boldsymbol{B}$, the optimal deployment is

$$\min_{\boldsymbol{r},\, \boldsymbol{l}} \mathrm{Risk}(\boldsymbol{r}, \boldsymbol{l})$$

subject to $\sum_{c\in C}\sum_{i\in r_c} R_i + \sum_{i\in \bigcup_c r_c} D_i + \sum_{i\in I} H_{l_i} \leq \boldsymbol{B}$

How to quantify security risks?

$$\text{Risk} = \sum_{\text{outcome}} \Pr[\text{outcome}] \cdot \mathit{Impact}(\text{outcome})$$

which components are compromised

what is the probability that they are compromised

what is the impact of their compromise on the system

Probability of Compromise

- Each implementation i is vulnerable with probability $1 - S_{l_i}$ (independently of other implementations)
- Instances of vulnerable implementations are compromised
- A component is compromised if

	Component Type			
	sensor	computational	actuator	interface
stealthy attack	**all** instances are compromised	**all** instances are compromised or **all** input components are compromised		
non-stealthy attack	**majority** of instances are compromised	either **majority** of instances are compromised or **majority** of input components are compromised		

Impact of Compromise

- Impact depends on the set of compromised components

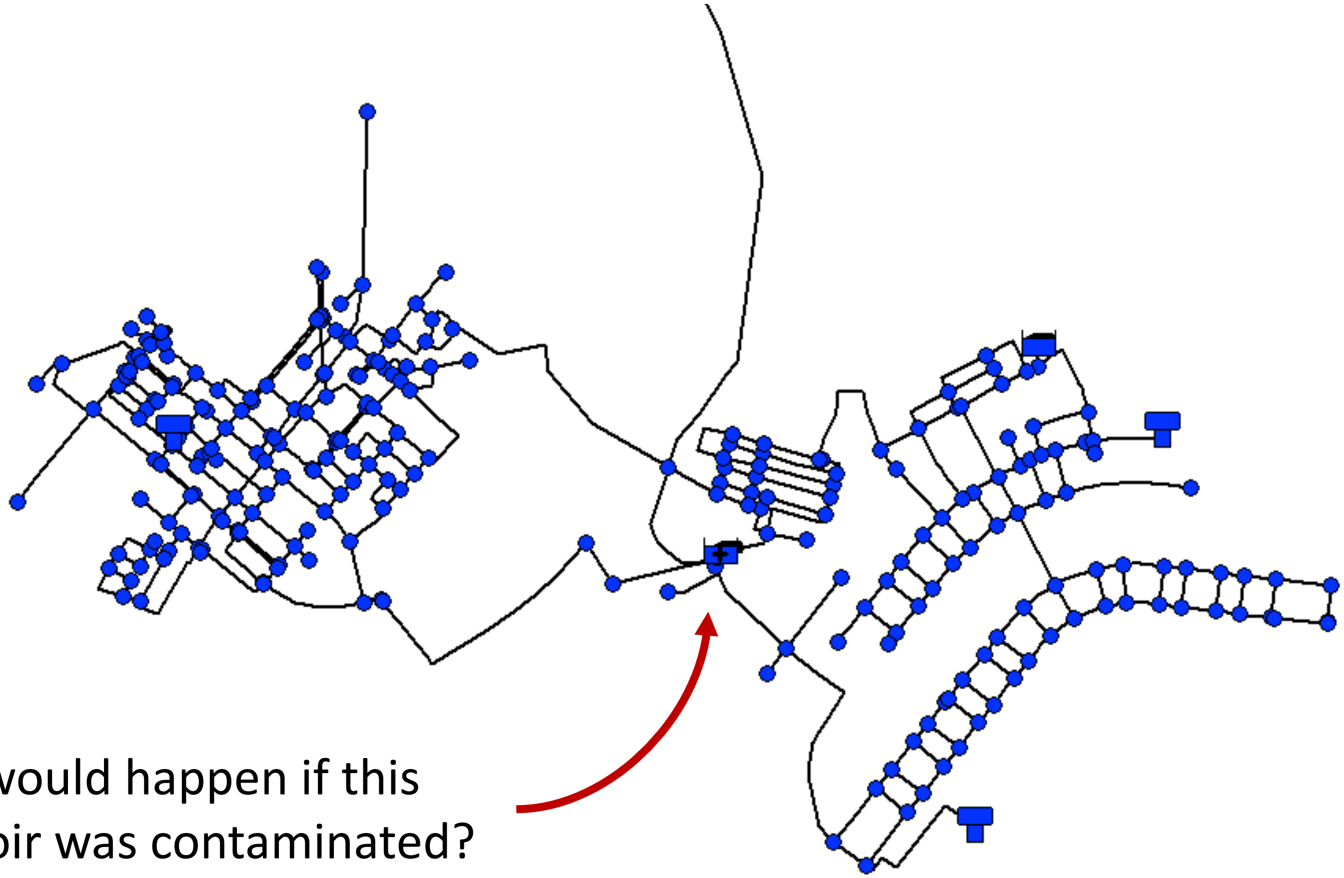
$$Impact = MaximumDamage(\text{compromised components})$$

  - exact formulation depends on the system

- We present two example systems
  1. smart water-distribution monitoring for contaminants
  2. transportation networks

Water-Distribution Networks

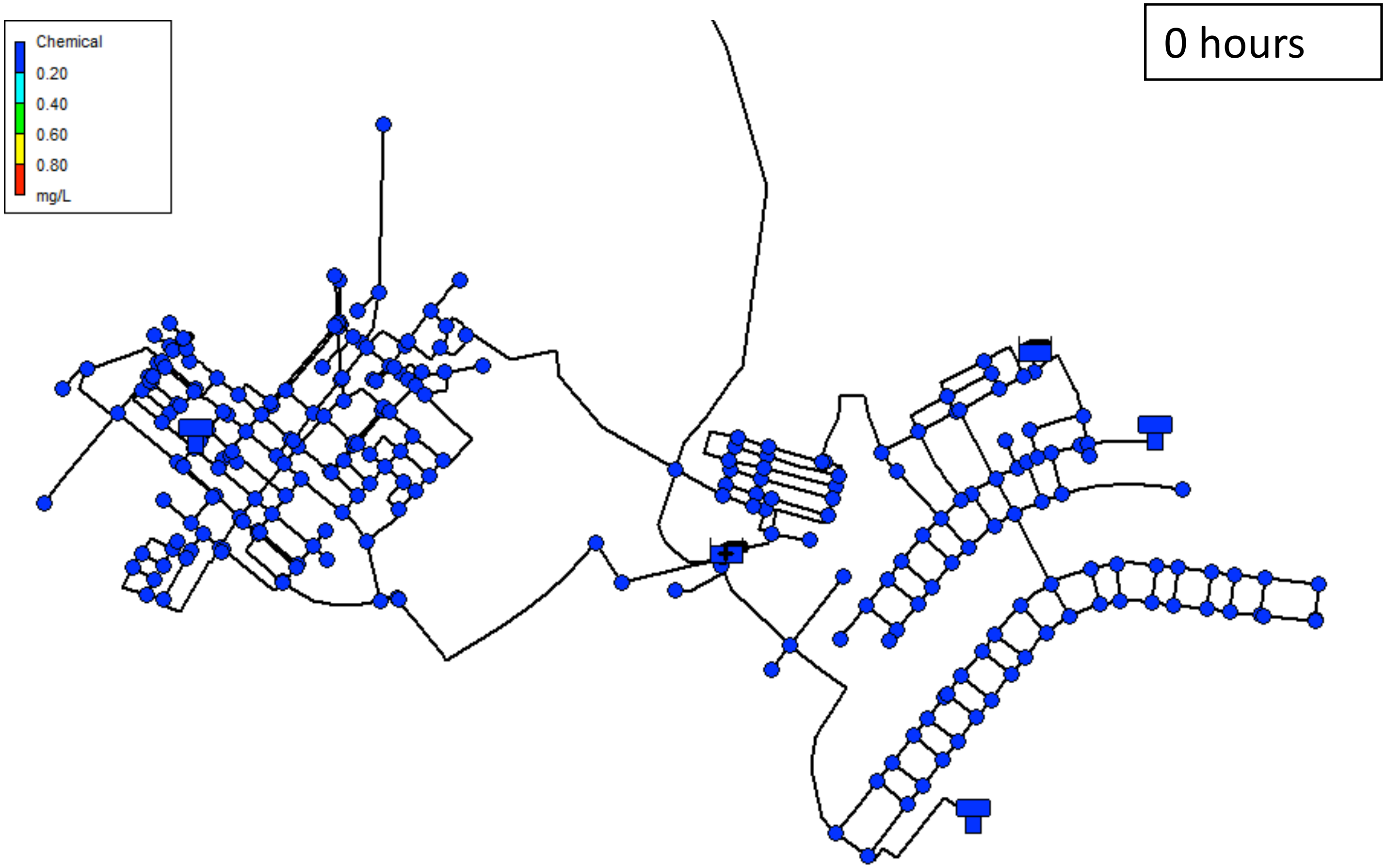
- Example topology (real residential network from Kentucky)



What would happen if this reservoir was contaminated?

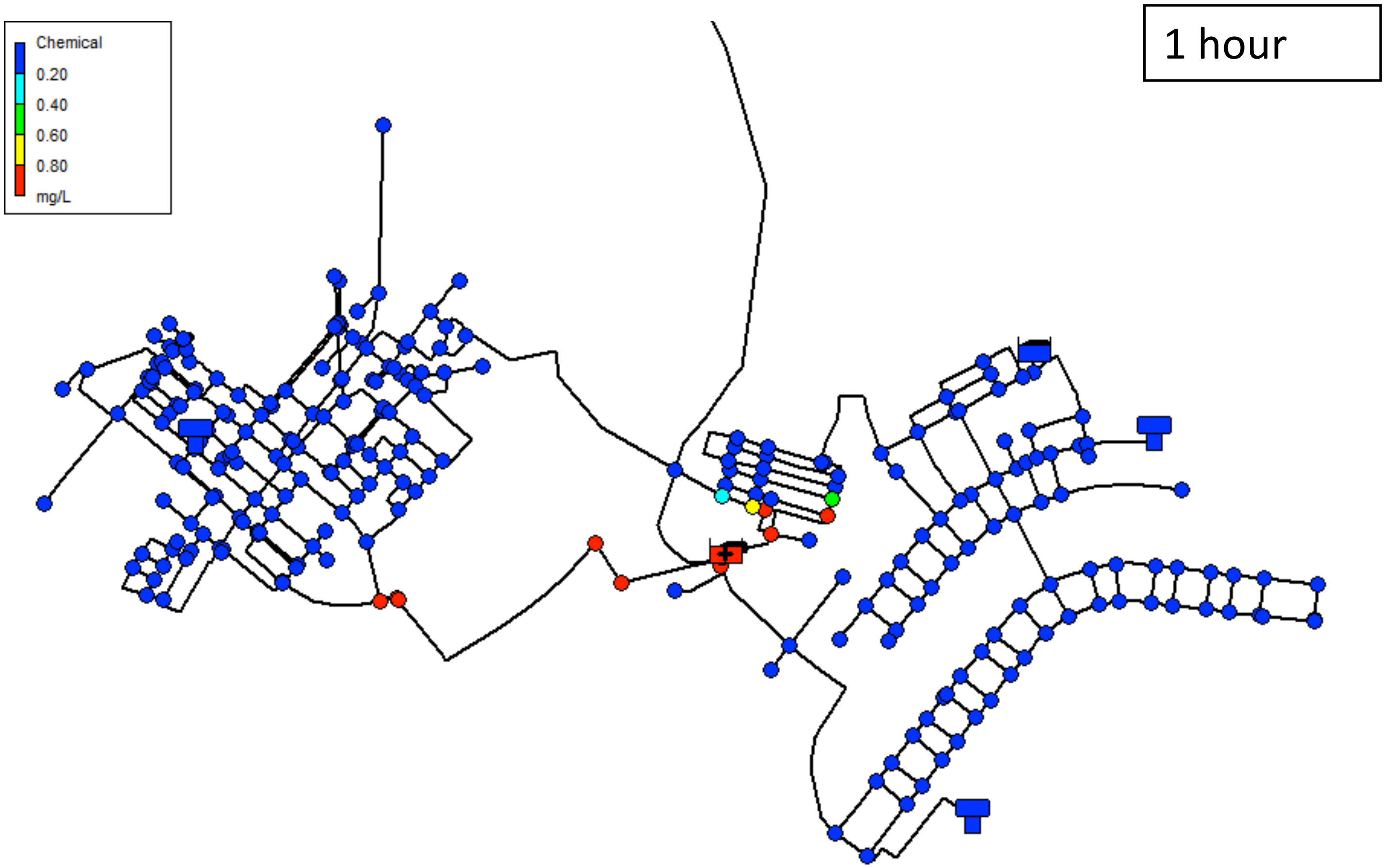
Contamination in Water-Distribution Networks

- Simulation using EPANET



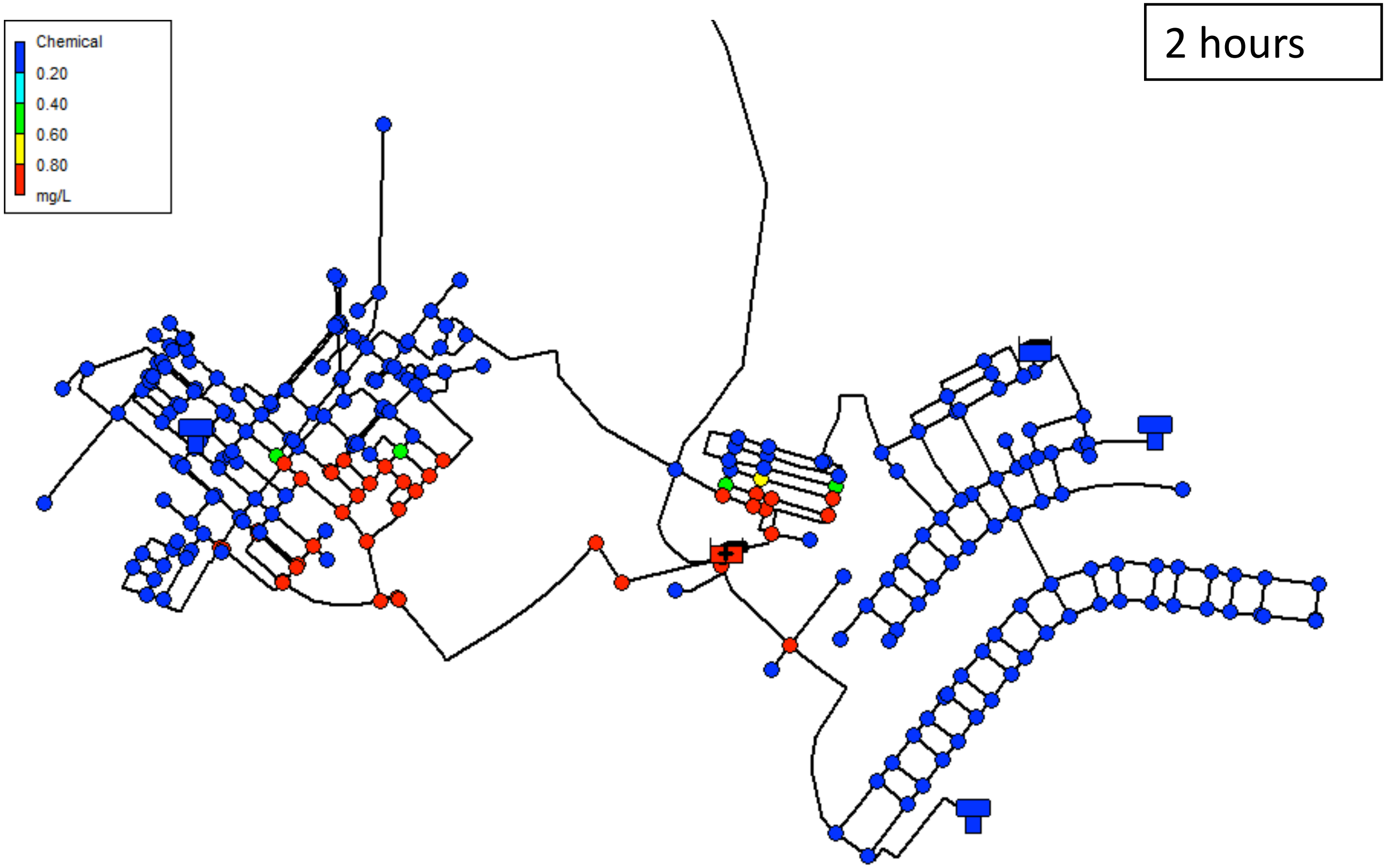
Contamination in Water-Distribution Networks

- Simulation using EPANET



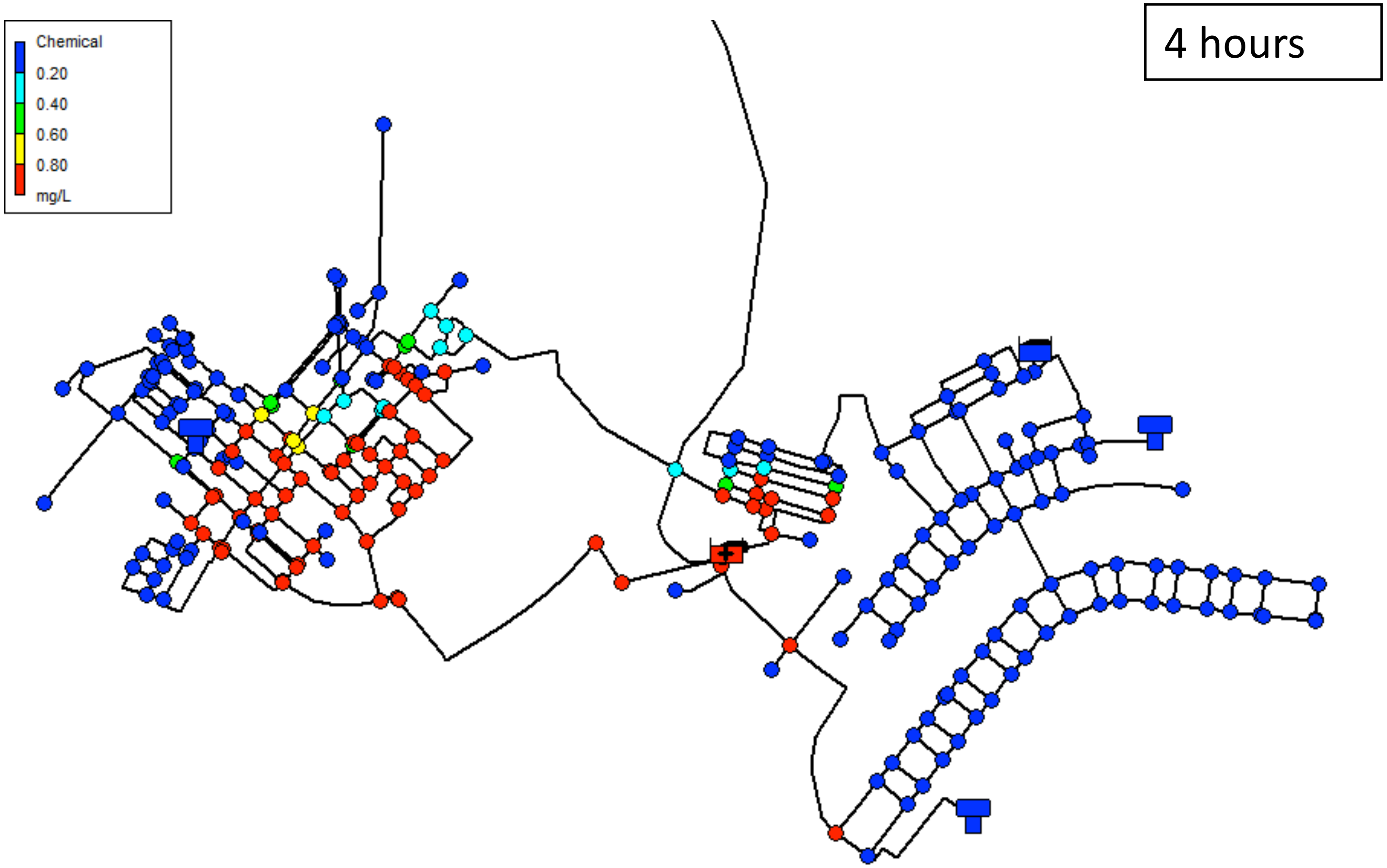
Contamination in Water-Distribution Networks

- Simulation using EPANET



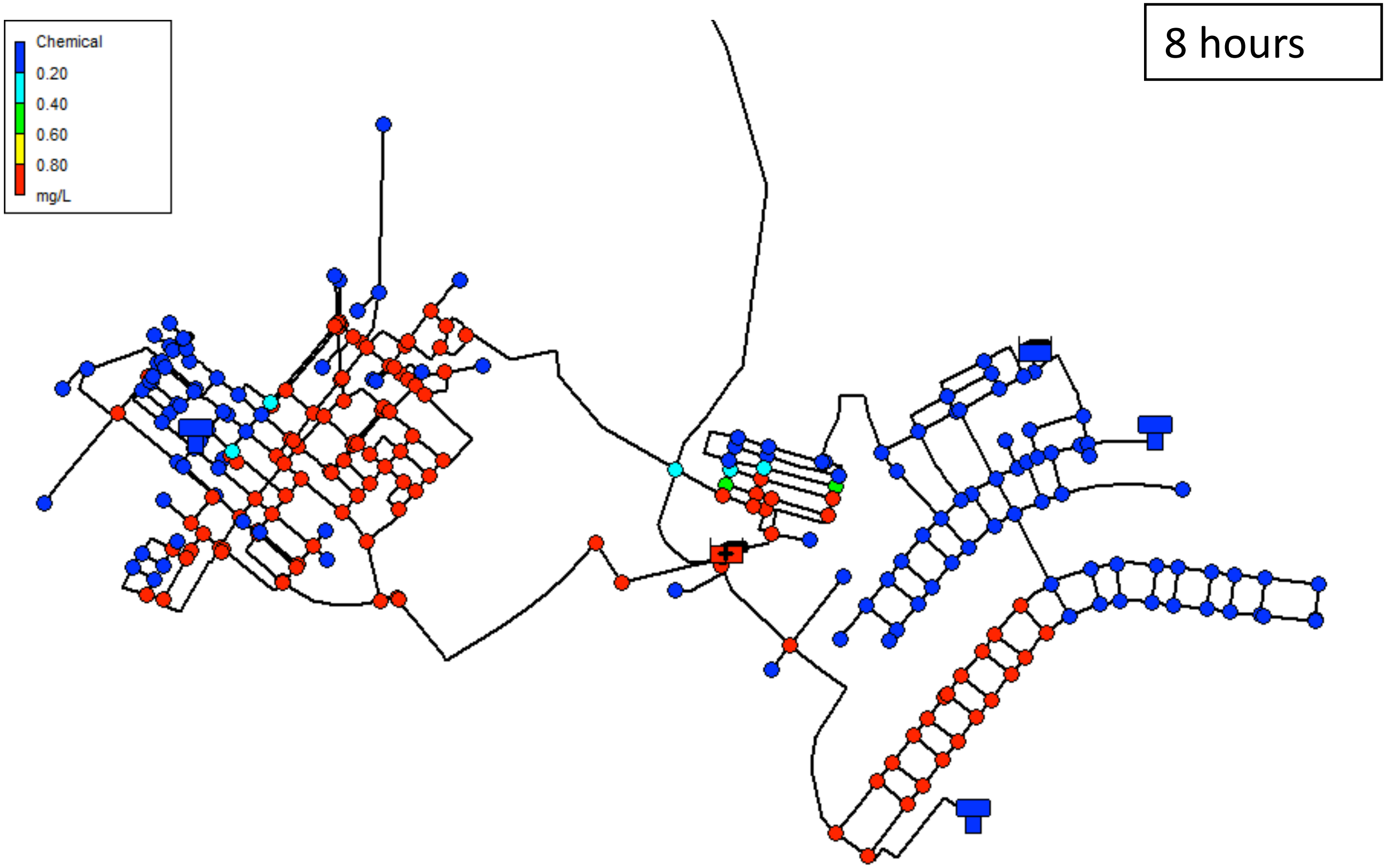
Contamination in Water-Distribution Networks

- Simulation using EPANET



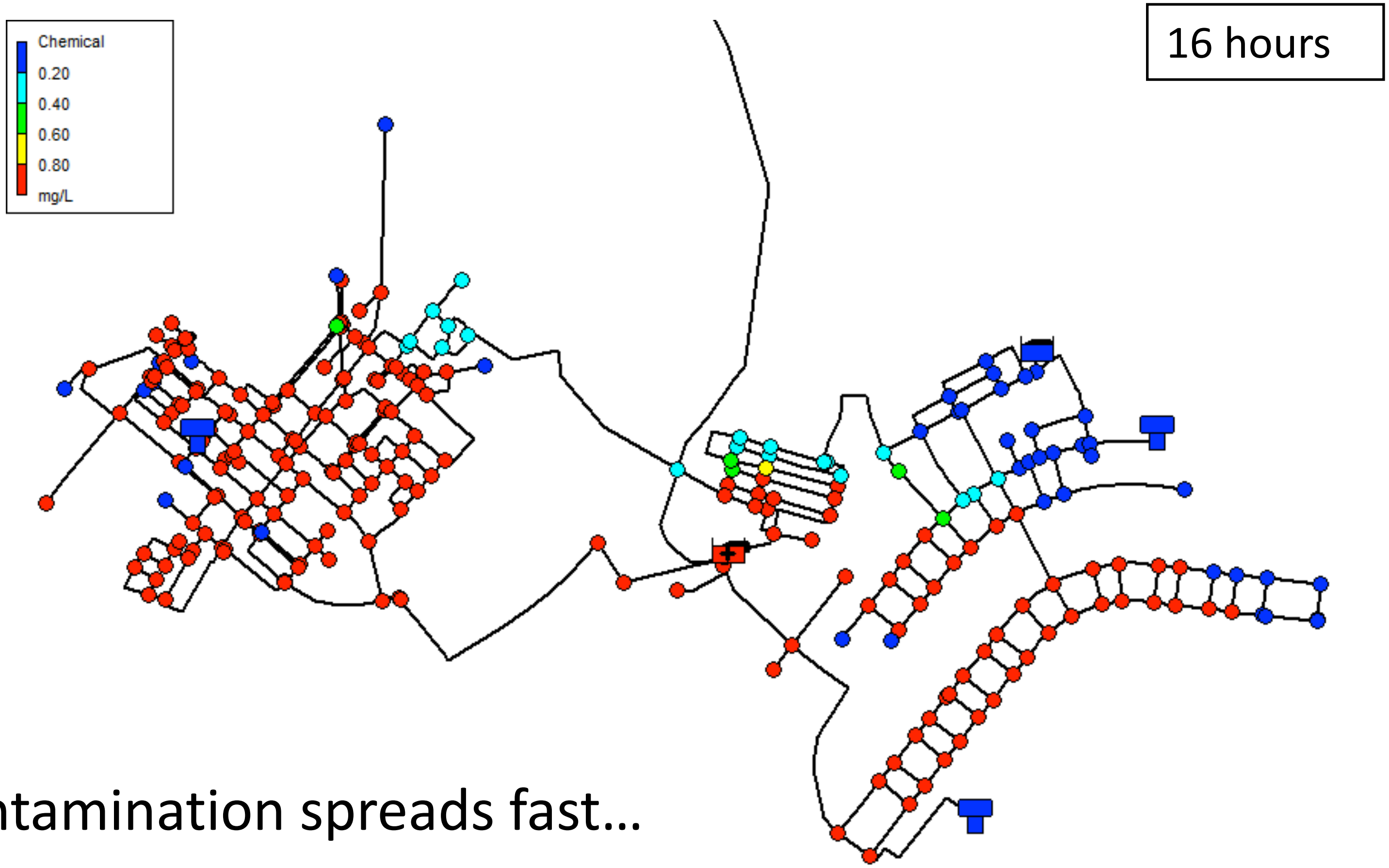
Contamination in Water-Distribution Networks

- Simulation using EPANET



Contamination in Water-Distribution Networks

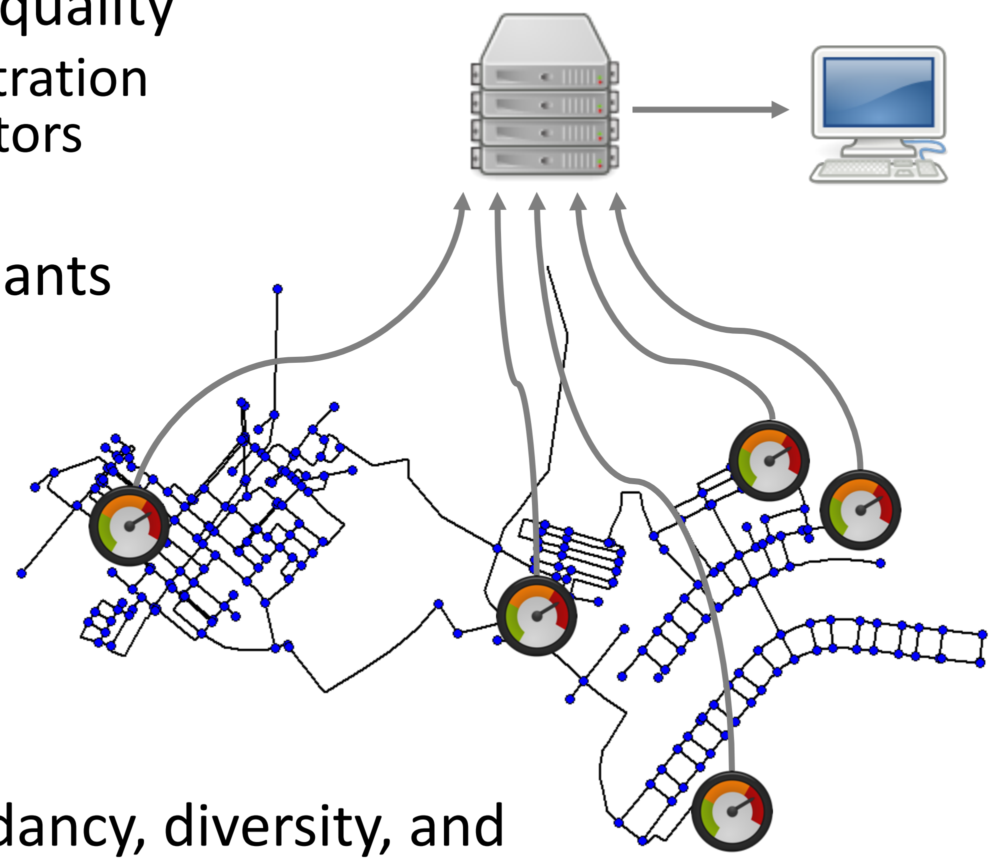
- Simulation using EPANET



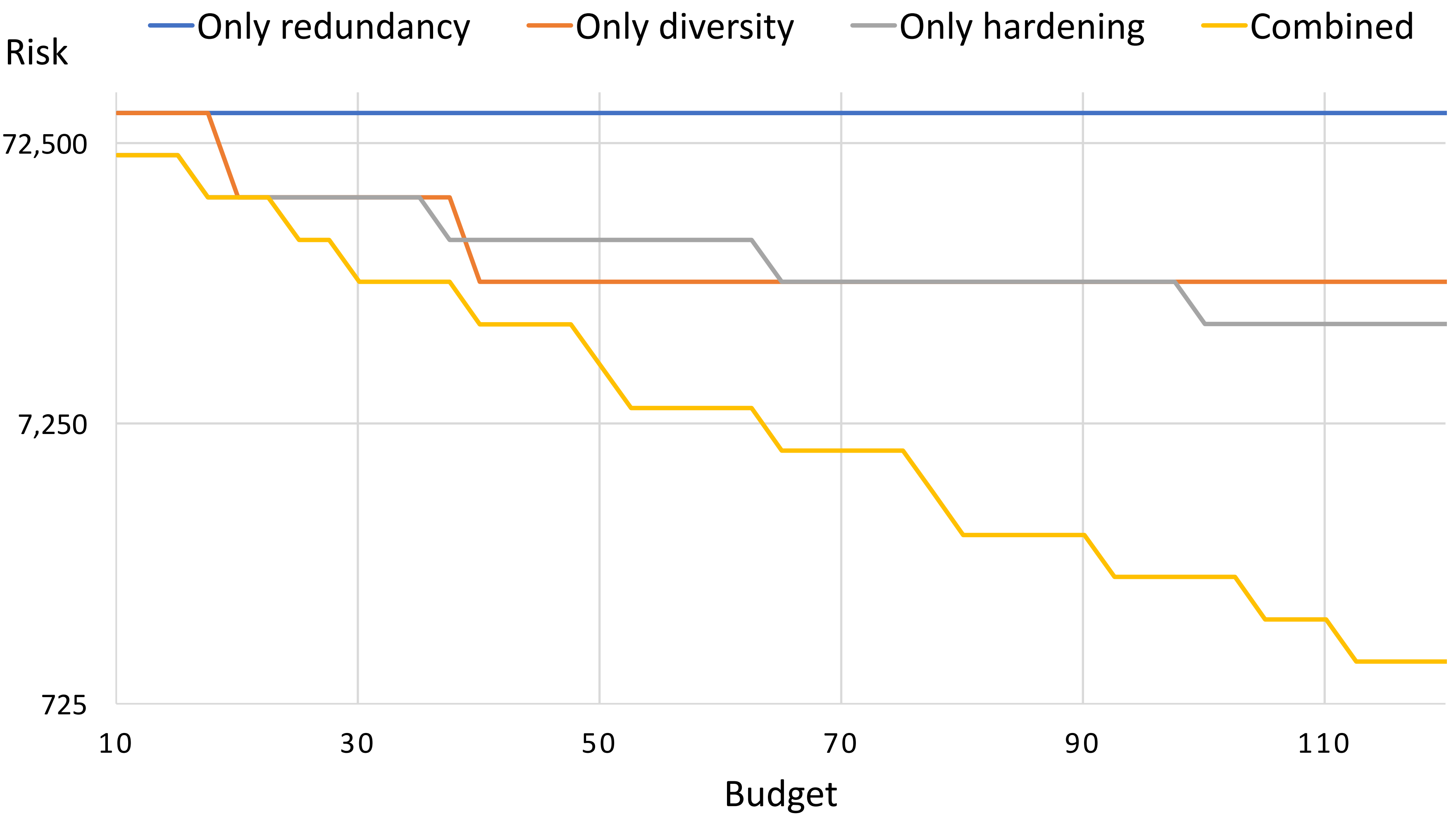
Contamination spreads fast…

Monitoring Water Quality

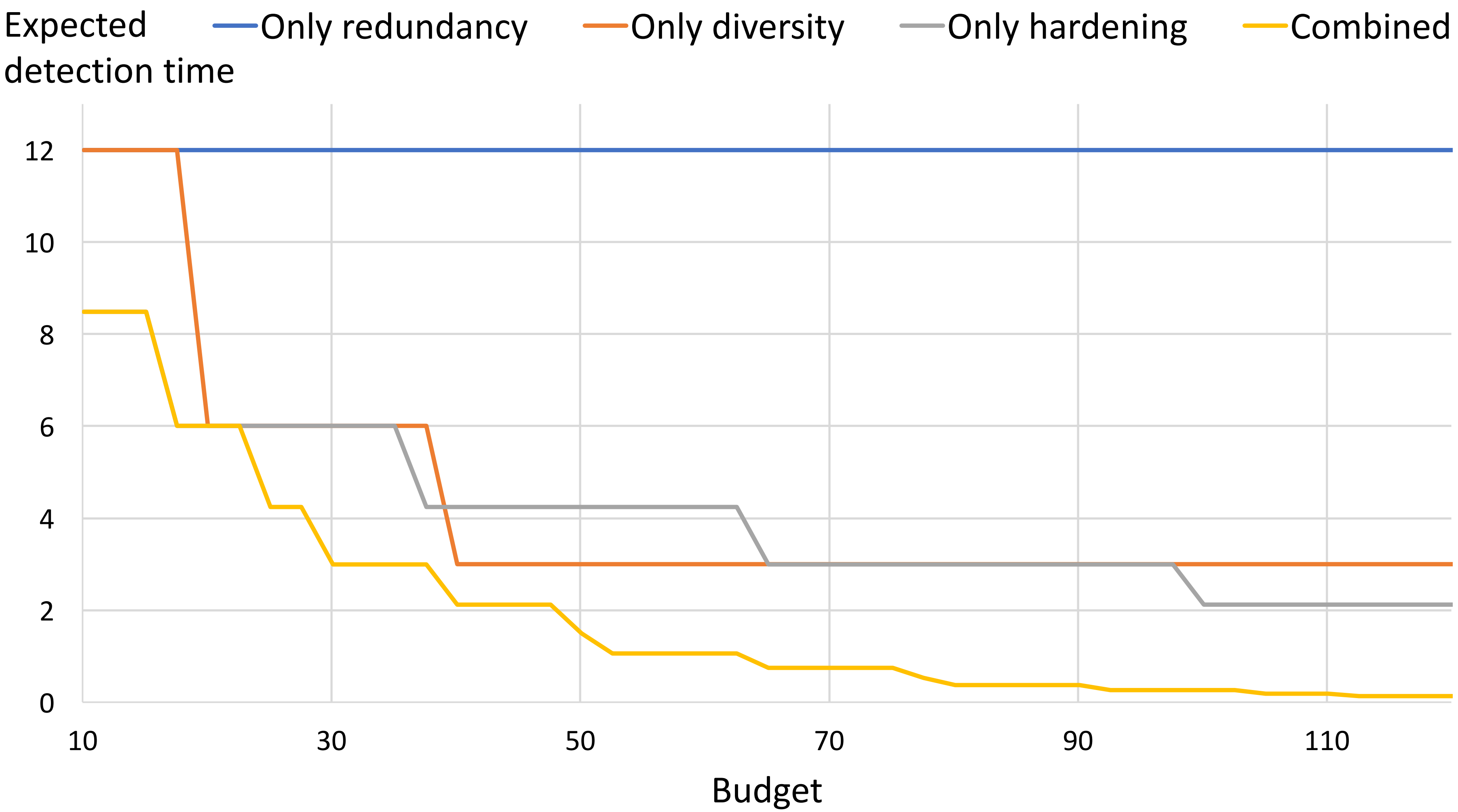
- We can deploy sensors that continuously monitor water quality
  - when contaminant concentration reaches a threshold, operators are alerted
- Impact: amount of contaminants consumed by the residents before detection
- Cyber-physical attack
  - compromises and disables vulnerable sensors
  - contaminates the reservoir that maximizes impact
- Defender invests into redundancy, diversity, and hardening for sensors



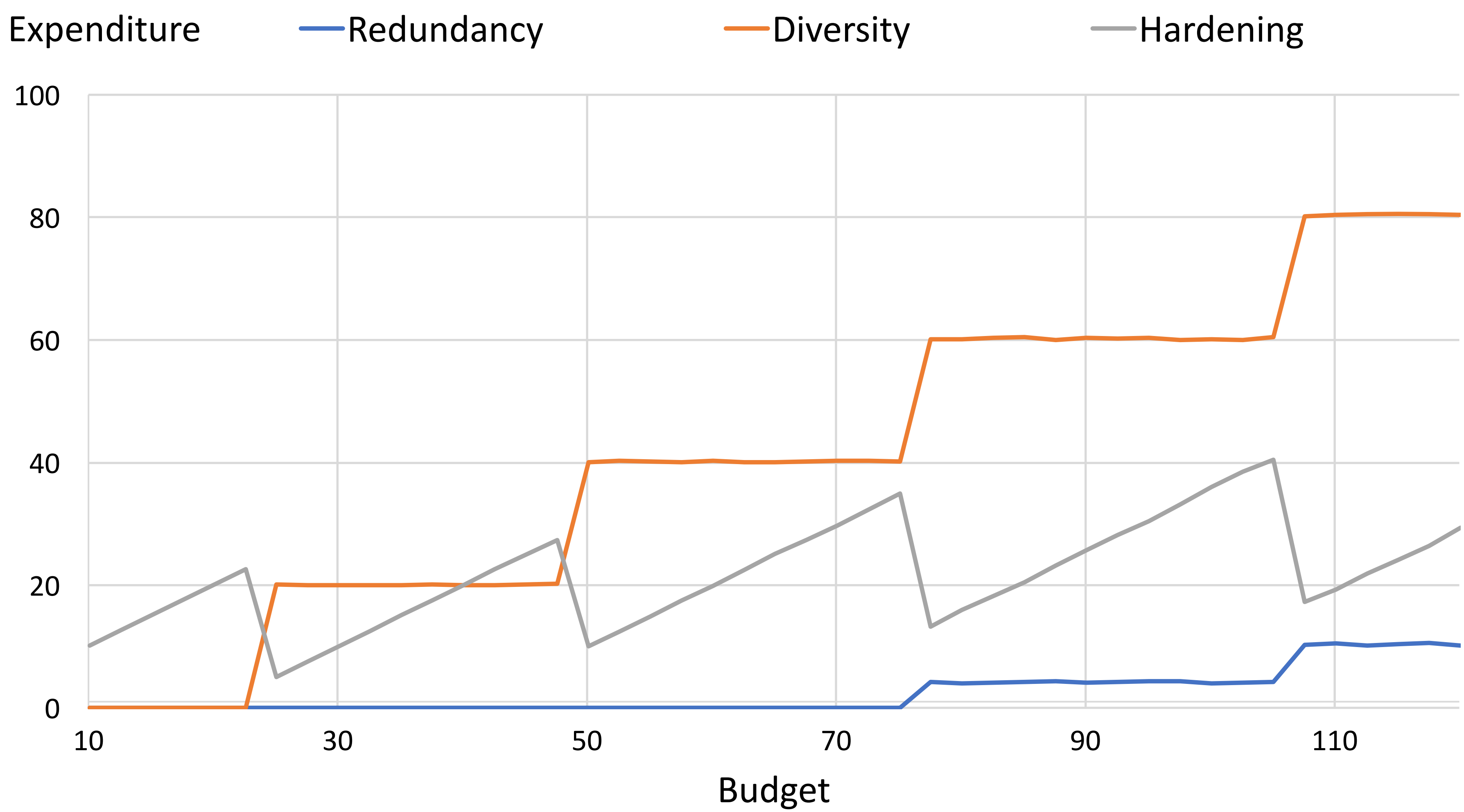
Security Risks



Expected Detection Time



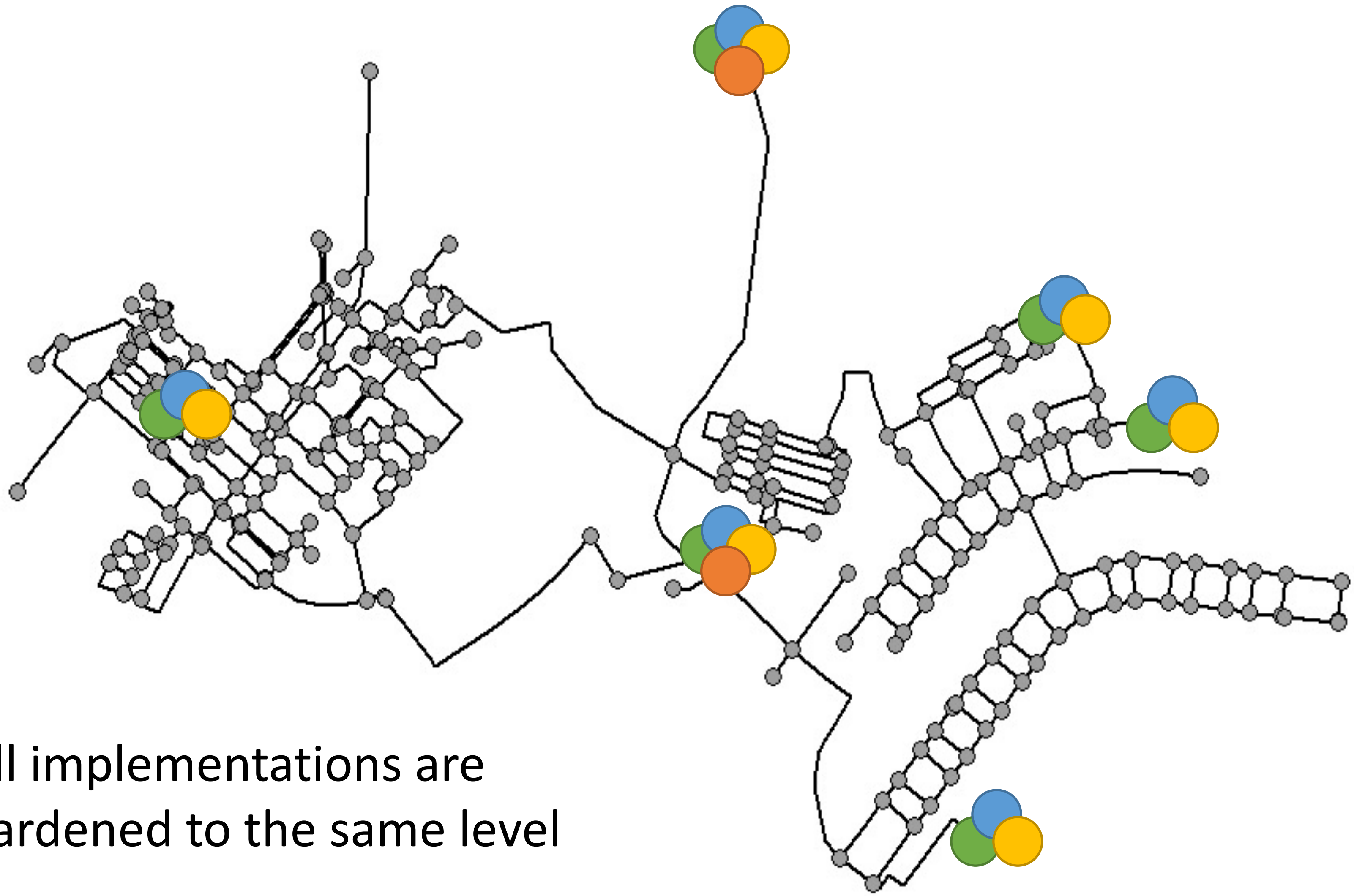
Optimal Allocation of Investments



Optimal Allocation of Investments

Budget	Redundancy	Diversity	Hardening
10	0	0	10
20	0	0	20
30	0	20	10
40	0	20	20
50	0	40	10
60	0	40.2	19.8
70	0	40.2	29.8
80	4	60	16
90	4	60.3	25.7
100	4	60	36
110	10.4	90.4	19.2
120	10.2	80.4	29.4

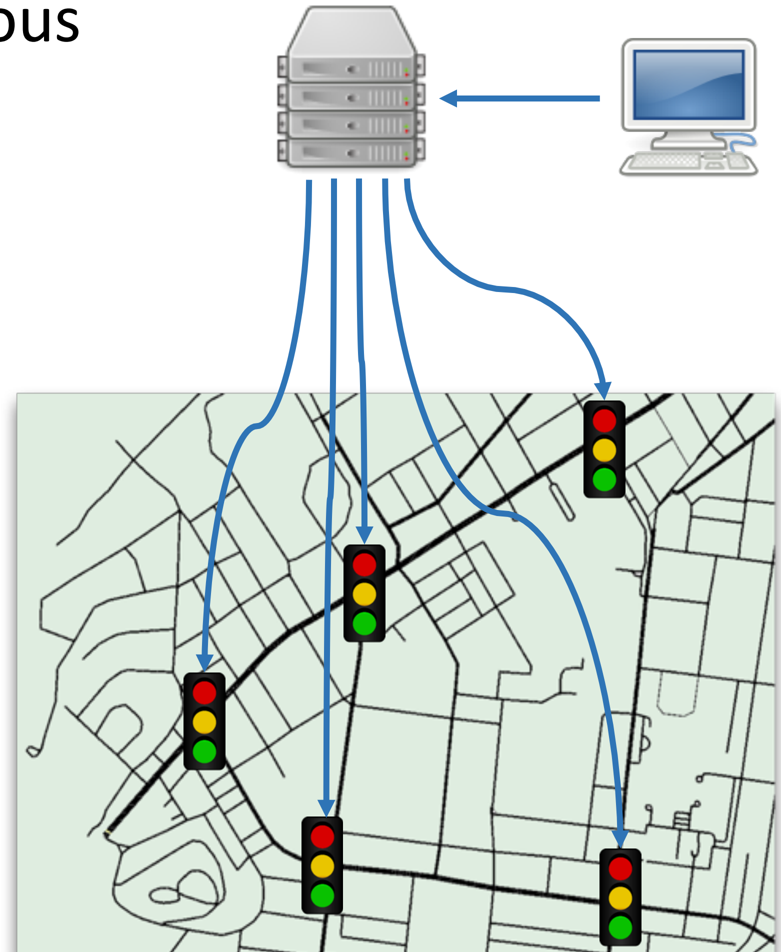
Optimal Deployment ($\boldsymbol{B} = 90$)



- All implementations are hardened to the same level

Transportation Network

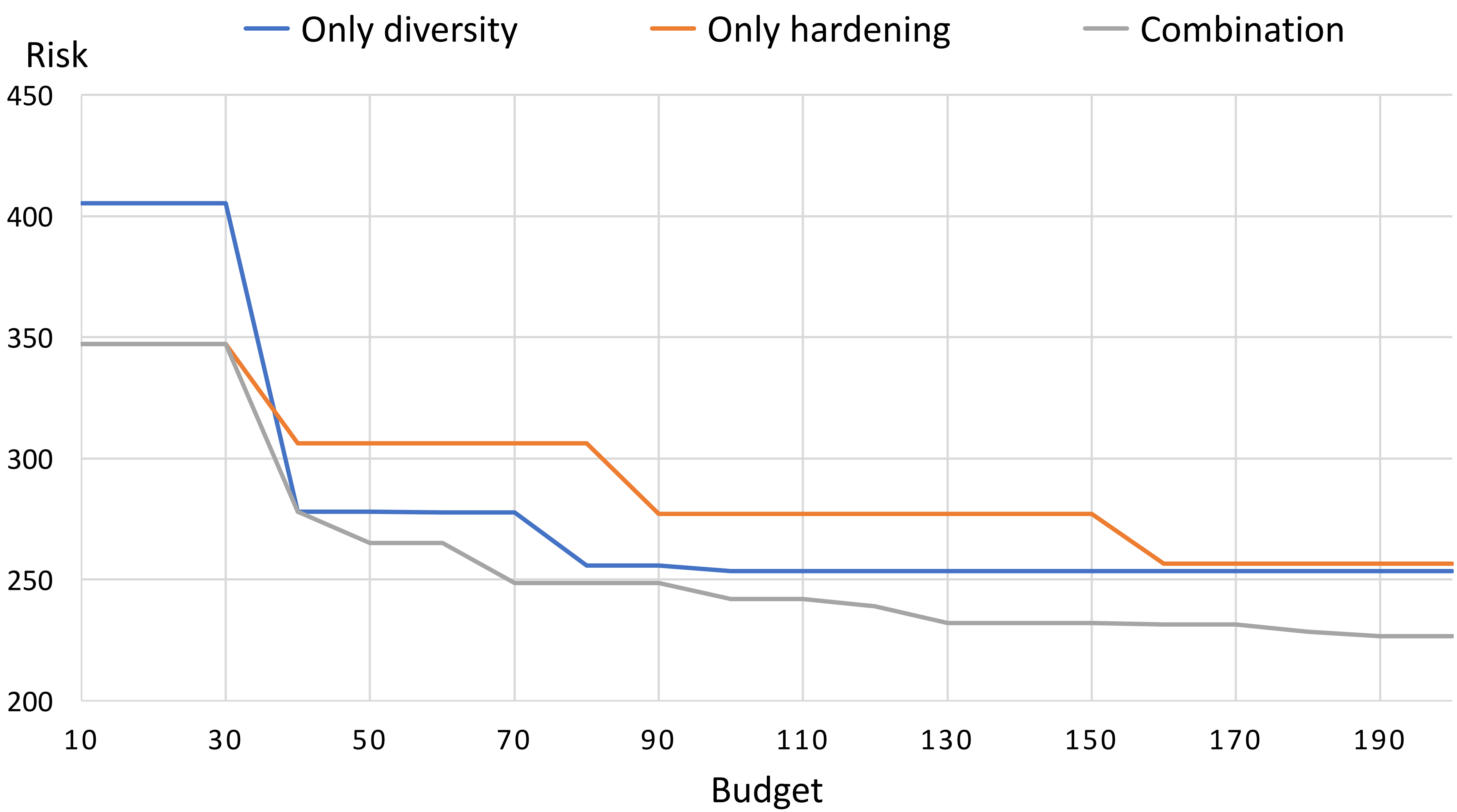
- Attacker may tamper with traffic control systems in order to cause disastrous traffic congestions
  - example:
    2006 incident in Los Angeles
- Component
  - embedded computer deployed at an intersection
  - controls the traffic lights
  - compromised components may be used by an attacker to disrupt traffic going through the intersection



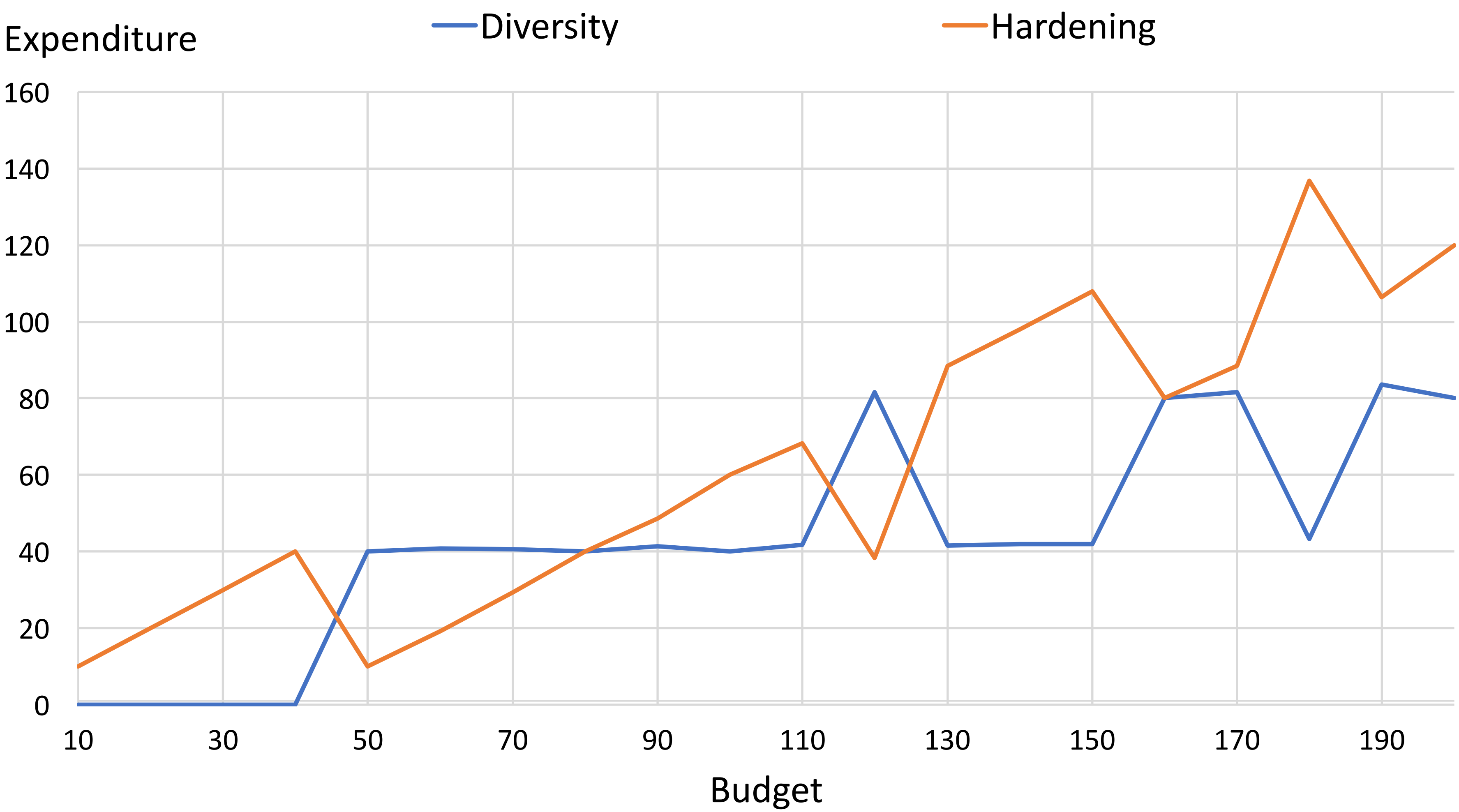
Transportation Network Risk Model

- We do **not consider redundancy** in this case since deploying redundant traffic light controllers requires additional assumptions
- Impact:
  increase in travel time
  due to adversarial tampering with traffic control
- Quantifying impact:
  traffic model
  - we use a well-known model, Daganzo's cell transmission model
  - compromised intersections are "blocked" (no through traffic)
  - travel time computed efficiently by solving the traffic model using a linear program

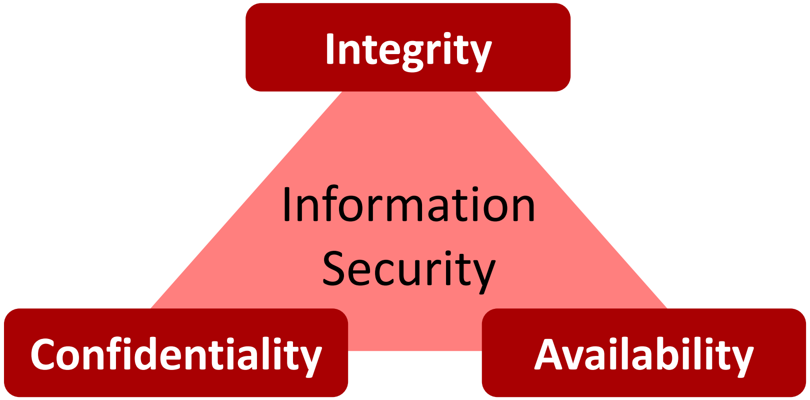
Security Risks

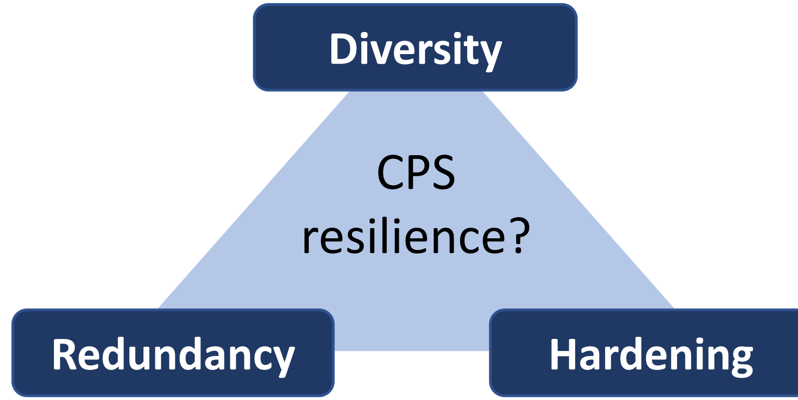


Optimal Allocation of Investment



Conclusion and Future Work





- There is no “silver bullet” approach for improving the robustness of cyber-physical systems
- The basic components of information security are confidentiality, integrity, and availability
- What are the basic components of CPS resilience?
- How do we organize, analyze, integrate, and evaluate the broad range of techniques that are available?

Thank you for your attention!

Questions?

Aron Laszka (alaszka@uh.edu)

Waseem Abbas (w.abbas@itu.edu.pk)

Yevgeniy Vorobeychik (yevgeniy.vorobeychik@vanderbilt.edu)

Xenofon Koutsoukos (xenofon.koutsoukos@vanderbilt.edu)