A Schedule Randomization Policy To Mitigate Timing Attacks in WirelessHART Networks

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Ankita Samaddar, Arvind Easwaran and Rui Tan

School of Computer Science and Engineering, Nanyang Technological University, Singapore

Background

i. **Components of a WirelessHART network**
   Sensors, actuators, gateway, network manager
ii. **Centralized architecture**
   Network manager generates schedules, allocates resources and decides routes
iii. **Complex mesh topology**

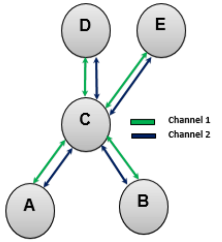
Characteristics of a WirelessHART network

- most reliable standard for real-time communication in time-critical systems
- Time Division Multiple Access (TDMA) based communication
- Channel diversity, route diversity and channel blacklisting
- Spatial re-use of channels

System Model

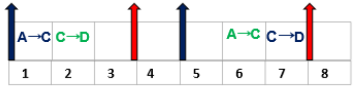
Our system consists of

- a network graph **G = (V , E)** with **m** channels
  V : set of nodes
  E : set of edges
- **n** end-to-end real-time flows
  F = {F_1,F_2,……F_n}



A **real-time flow** in a WSN in CPS is defined as

F = {s,d,p,δ,R}
s: source of a flow
d: destination of a flow
p: period of a flow
δ: deadline of a flow
R: set of routes from s to d
F_1 = {A, D, 4, 3, [A→ C→D]}



Motivation

Scheduling policy in a WirelessHART network is

1. Centralized
2. The same communication schedule repeats over every hyperperiod (L.C.M. of the periods of all the flows)
3. The time slots in a schedule are predictable in nature

Consequence :

- Due to the repetitive nature of the communication schedule over every hyperperiod, the attacker can ***predict*** the time slots in which any two target device communicate
- The attacker can launch ***selective jamming attacks*** targeting specific transmissions

Countermeasure to Attack

We propose ***a schedule randomization policy***, the ***SlotSwapper***, as a countermeasure to attack

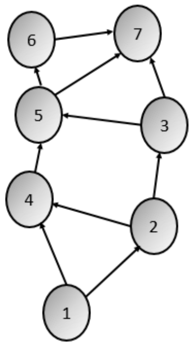
SlotSwapper consists of two phases –

- an **offline randomized Schedule Generation Phase** (runs at the network manager)
- an **online Schedule Selection Phase** (runs at each node in the network)

Offline Schedule Generation Phase

Step 1: Consider a **base schedule B** over a set of three flows

F = {s, d, p, δ, R}
F_1 = {1, 7, 8, 8, [1 → 2 → 3 → 7] }, F_2 = {4, 7, 4, 4, [4 → 5 → 7] },
F_3 = {2, 7, 8, 8, [2 → 3 → 7] }



Base Schedule B

	1	2	3	4	5	6	7	8
Chan 1	1→2		5→7	2→3	4→5			3→7
Chan 2	4→5				2→3	3→7	5→7	

Step 2: Consider the **scheduling window [1,7]** for the first hop **(2→3)** of **F_3**
Step 3: Check for **transmission conflicts, deadline preservation** and **flow sequence preservation** within scheduling window **[1,7]**
Step 4: Generate an **eligible list of slot-channel pairs** that satisfy all the conditions from **Step 3**

	1	2	3	4	5	6	7	8
Chan 1	1→2		5→7	2→3	4→5			3→7
Chan 2	4→5				2→3	3→7	5→7	

Step 5: Select a **random slot-channel pair** from the list of eligible slot-channel pairs
Step 6: Swap the current slot with the randomly selected slot

	1	2	3	4	5	6	7	8
Chan 1	2→3		5→7	1→2	4→5			3→7
Chan 2	4→5				2→3	3→7	5→7	

Step 7: Repeat steps 2 to 6 for every hop of each flow instances in B
Run the offline Schedule Generator for a large number of times to generate a set of feasible randomized schedules

Online Schedule Selection Phase

Each node selects a schedule uniformly at random from the set of feasible schedules at every hyperperiod
All nodes in the network select the same schedule independently
- Each node uses the same pseudo random number generator initialized with the same seed

Base Schedule B

	1	2	3	4	5	6	7	8
Chan 1	1→2		5→7	2→3	4→5			3→7
Chan 2	4→5				2→3	3→7	5→7	

Randomized Schedule S after offline Schedule Generation Phase

	1	2	3	4	5	6	7	8
Chan 1		1→2		5→7	3→7			5→7
Chan 2	2→3	4→5	2→3			4→5	3→7	

Measure of Uncertainty – Kullback Leibler Divergence

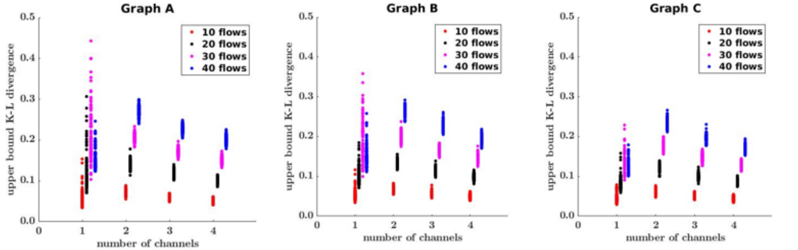
We propose **Kullback Leibler Divergence or K-L Divergence** as a **security metric** to compare the performance of our algorithm w.r.t. a truly random algorithm

- Measures the **divergence** between the probability distribution of the flows in the schedules generated by two algorithms

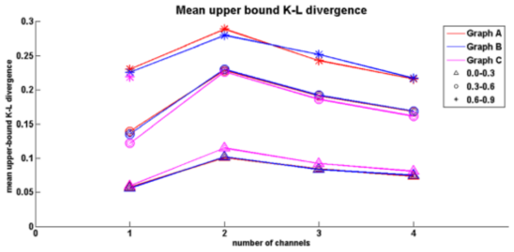
$$\mathcal{D}(\mathbb{A}||\mathbb{A}') = \sum_{i=1}^{l}\sum_{k=1}^{m}\sum_{j=0}^{n} \Pr(g_{ik}=j)\log_2\frac{\Pr(g_{ik}=j)}{\Pr(g'_{ik}=j)}$$

where **Pr($g_{ik=j}$)** and **Pr($g'_{ik=j}$)** are the probability mass functions of the j^{th} flow occurring in the k^{th} channel of the i^{th} slot by algorithm A and A' respectively

Experimental Results



Upper-bound K-L divergence over randomly generated sparse, medium and dense graphs with number of flows varying between 10 to 40 and number of channels between 1 to 4 over a hyperperiod of 1024 slots



Mean upper-bound K-L divergence over randomly generated sparse, medium and dense graphs with utilization varying between 0.0 to 0.9 and number of channels between 1 to 4