Trust Trackers for Computation Offloading in Edge-Based IoT Networks

WARWICK

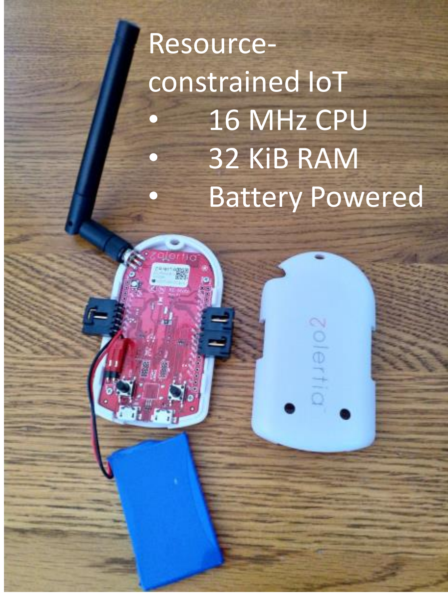
Matthew Bradbury, Arshad Jhumka and Tim Watson

2:00 – 3:30 PM EDT, 11th May 2021

INFOCOM 2021

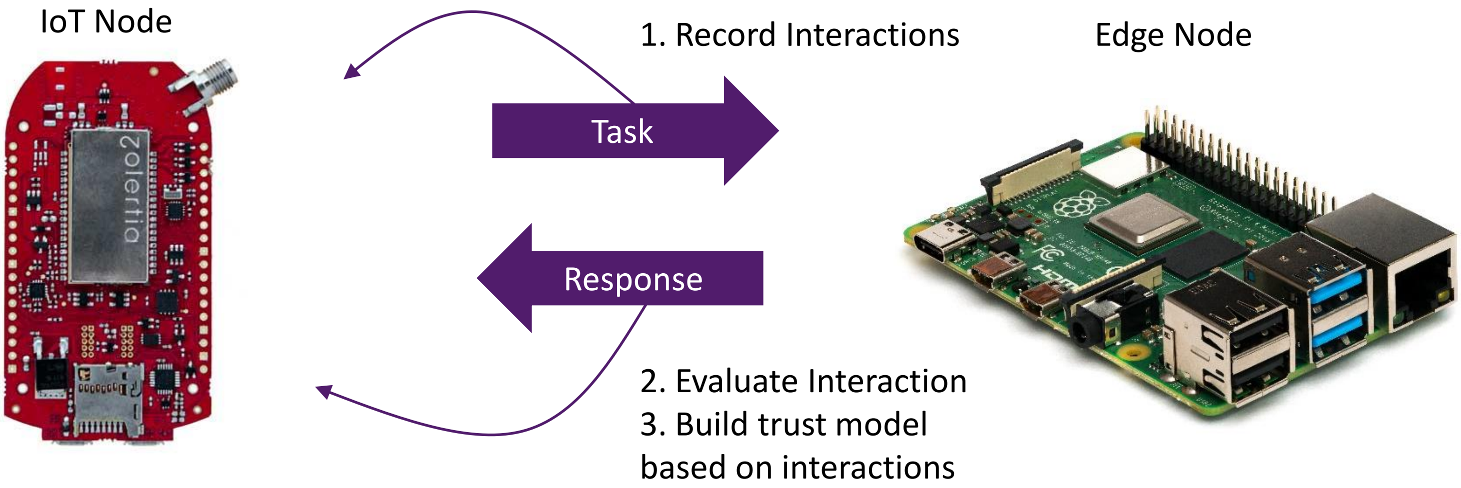
Introduction

- Wireless IoT devices are useful for deployment when physical access to infrastructure is restricted (costly, untrusted, unavailable)
- These devices are constrained (limited CPU, RAM, data storage) to maximise lifetime when battery powered
- These devices will have expensive tasks that they need to perform
- As the devices are constrained, expensive tasks can be offloaded to Edge nodes with greater capabilities
- Which Edge node is chosen for these tasks to offload?



Trust Assessment

- Use a measure of *behavioural trust* to assess which Edge is most likely to perform well
- Typically assessed *reactively* based on past events
- Instead, this work investigates *proactive* trust assessment



This Talk

1. Formalise the offloading problem
2. Prove:
    1. It cannot be solved in an asynchronous network
    2. It can be solved by a trust tracker device in synchronous networks
    3. That the trust tracker device cannot be implemented
3. Probabilistic offloading
4. Evaluate experimental results from a small (6 node) testbed

Offloading Problem

Note: permanently does not mean forever here, but long enough for the system to make progress

- For an IoT node, there exists an Edge node such that:
- Correctness: The IoT node offloads to the Edge node only if it trusts the Edge node
- Trust: Eventually, the IoT node trusts the Edge node permanently

An IoT node trusts an Edge node if it expects it to:

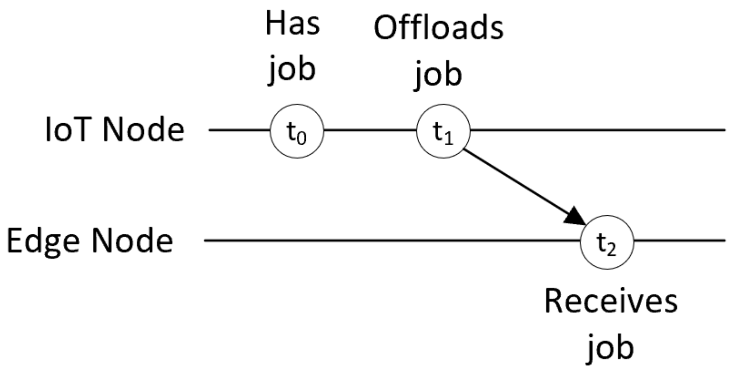
1. Acknowledge submitted tasks
2. Deliver a correct result
3. The result is delivered within a deadline

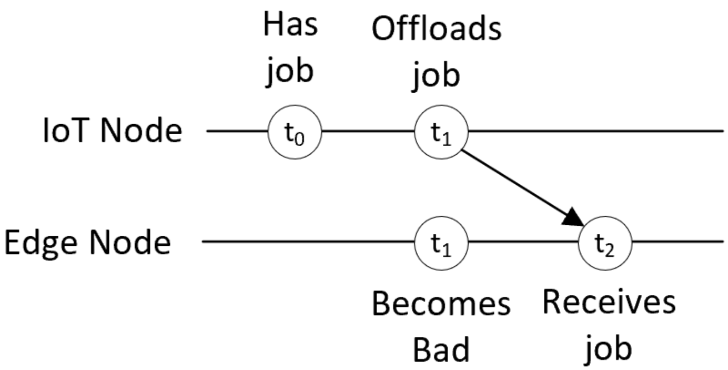
Offloading Engine (O)

- There is a software device that is responsible for offloading
- Safety: O returns a set of trusted nodes
- Liveness: Eventually, O returns a set of Edge nodes
- There might not be any good Edge nodes, so can’t expect a non-empty set!

Impossibility of Correct Offloading in an Asynchronous Network

- Asynchronous network = no bounds on time to perform computation or communication
- Edge node can become bad at the same time an IoT node decides to offload to it

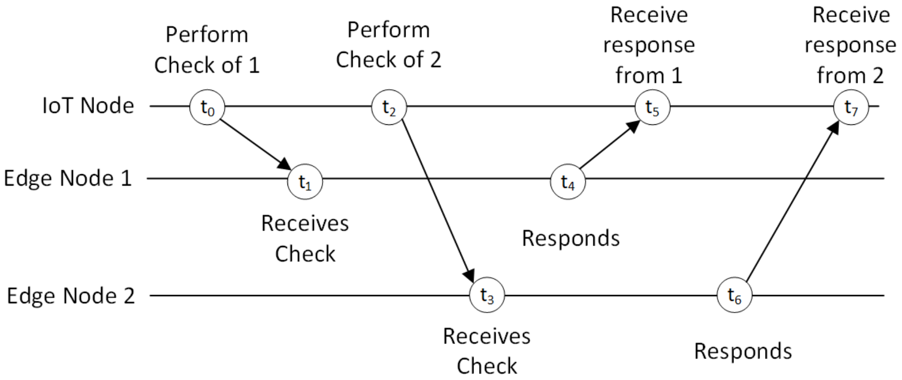


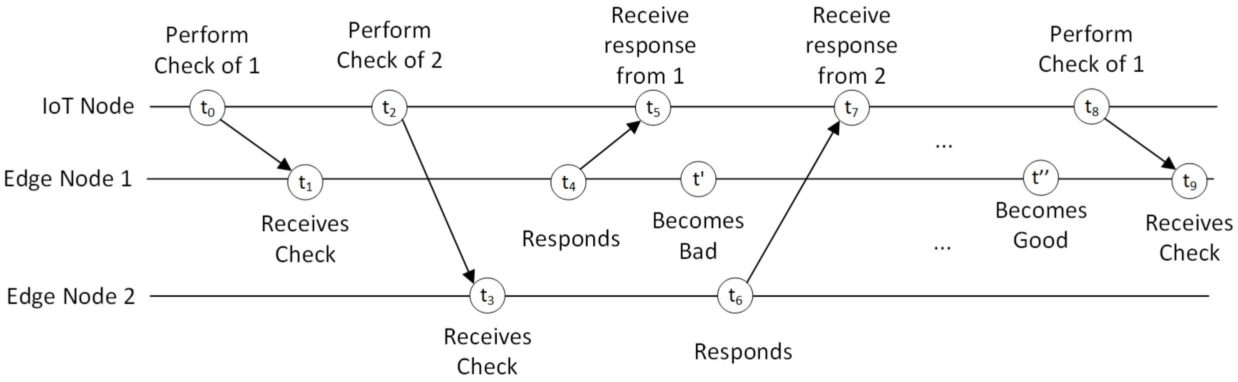


Trust Tracker Device (Σ) for Synchronous Networks

- Maintain an *epoch number*, that is incremented every time a change in behaviour occurs (bad → good or good → bad)
- Change in behaviour assess by a *challenge*
- Completeness: All bad Edge nodes are eventually suspected by all IoT nodes, or the epoch number is unbounded
- Accuracy: For some Edge nodes, all IoT nodes eventually permanently trust those Edge nodes and their epoch number stops changing
- O and Σ are equivalent
    - Test trust via the challenge
    - If there are any well-behaved Edges, will eventually identify them

Impossibility of Implementing the Trust Tracker Device





- Two runs, one with no failures and one with, both return the same result – that all Edge nodes are trusted

Probabilistic Offloading

- Cannot deterministically determine trustworthy behaviour
- Correctness: IoT node only offloads to an Edge if it trusts the Edge with high probability
- Trust: Eventually, the IoT node permanently trust the Edge with high probability

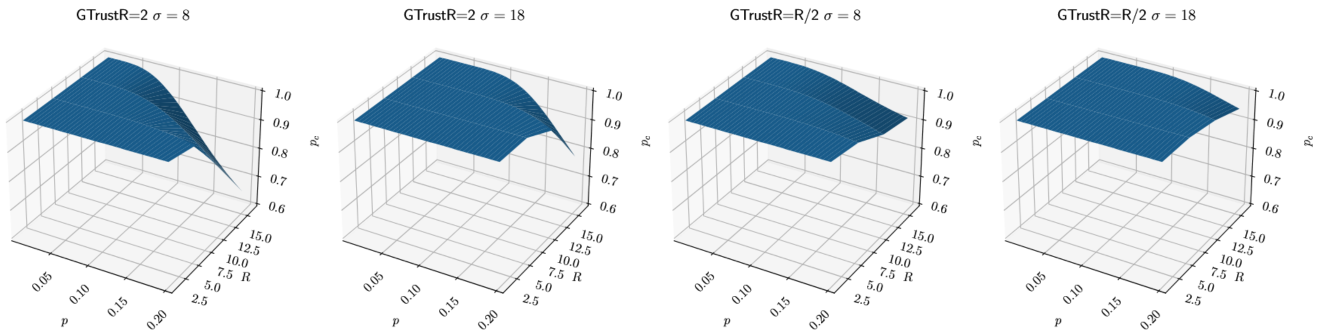
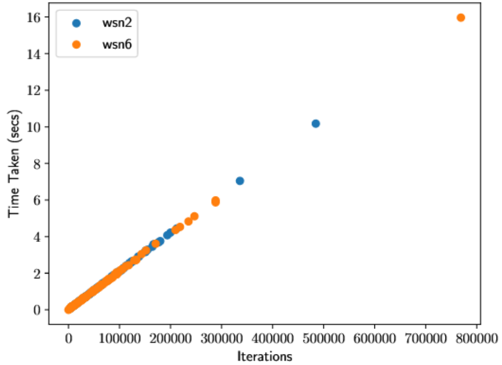


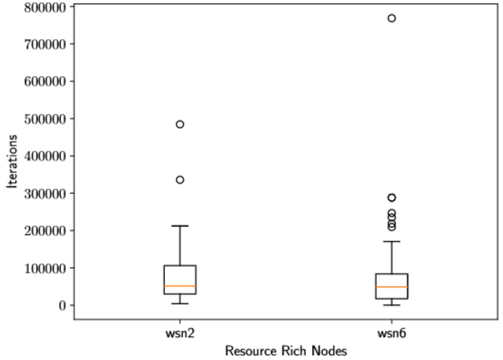
Figure 2: The probability of a correct offload (p_c) when varying: the number of resource-rich nodes (R), the probability of a resource-rich node being fake (p), the number of samples performed (σ), and the number of trustworthy nodes ($|GTrustR|$).

Proactive Trust Assessment

- IoT nodes periodically send a challenge to Edge nodes testing their behaviour
- Idea: If Edge nodes are willing to dedicate resources to an expensive challenge, they will be willing to do an expensive job
- Borrowed proof-of-work from blockchain as the Zolertia RE-Motes have hardware acceleration for SHA256
    1. IoT generates random 32 bytes b, difficulty d and a deadline t, send to Edge node
    2. Edge node finds a prefix to b such that the first d bytes of SHA256(p||b) are 0
- Consider: This does not assess Edge's ability to correctly execute tasks



(a) The number of prefixes searched to find a solution versus the time taken.

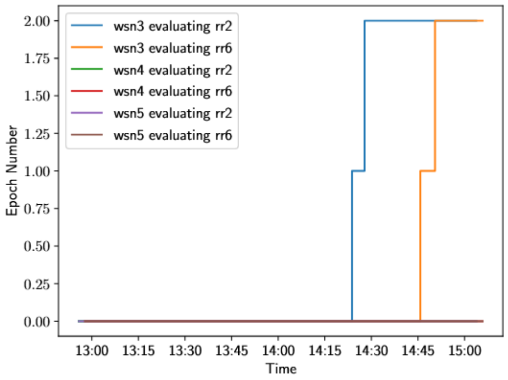


(b) A comparison between the load caused by the challenge on two different resource-rich nodes.

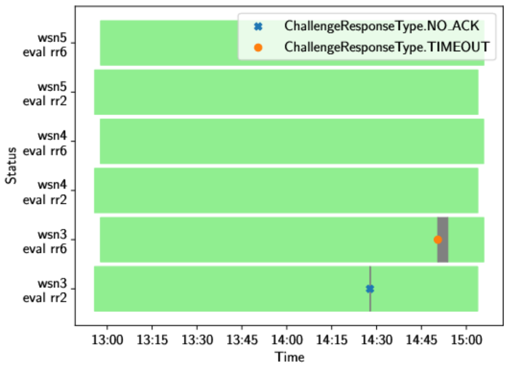
Figure 3: Challenge performance when both resource-rich nodes are good.

Challenge Overhead on Edge Nodes

- The challenge should be expensive to compute and not take too long
- A balance needs to be found between the cost of the challenge and resources dedicated to executing tasks
- Also (somewhat) important that there is no bias in which Edge nodes receive harder challenges

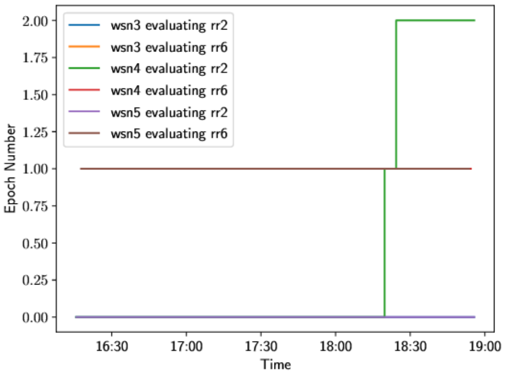


(a) Evolution of the Epoch number over time.

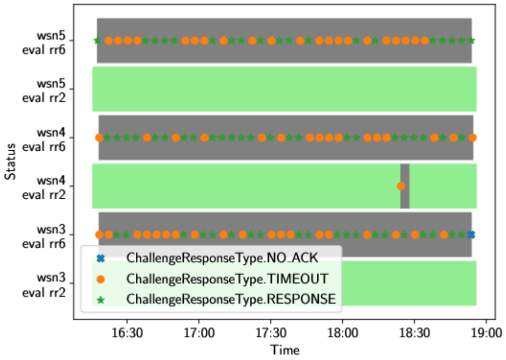


(b) Times at which resource-constrained nodes trusted resource-rich node. Events that led to loss of trust are indicated.

Figure 4: Results for when both resource-rich nodes 2 and 6 are good.



(a) Evolution of the Epoch number over time.

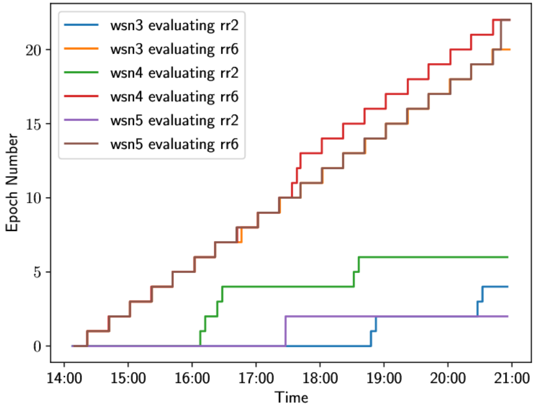


(b) Times at which resource-constrained nodes trusted a resource-rich node. Events that led to loss of trust are indicated.

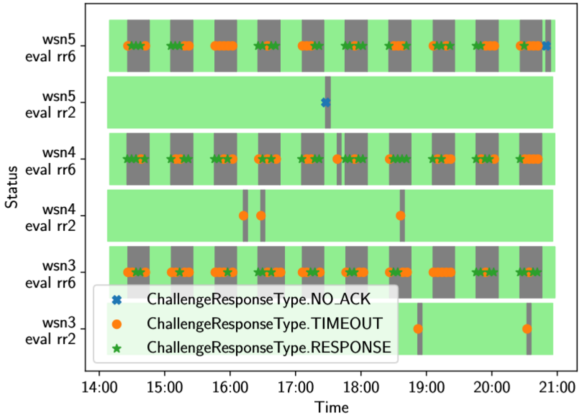
Figure 5: Results for when resource-rich node 2 is good and 6 is bad.

Stable Behaviour

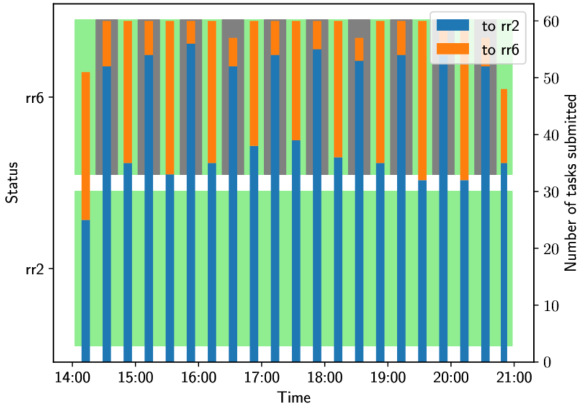
- Two experiments
  - Both edge nodes always good
  - One edge node (rr2) always good, the other (rr6) always bad



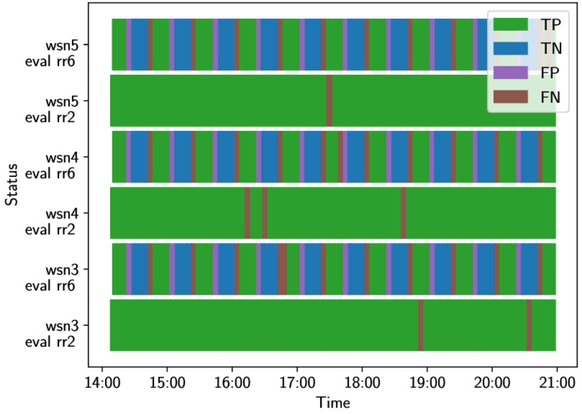
(a) Evolution of the Epoch number over time.



(b) Times at which resource-constrained nodes trusted resource-rich nodes. Events that led to loss of trust are indicated.



(c) The true status of resource-rich nodes and the number of tasks submitted to them in a time window where their behaviour was stable.



(d) Was the trust correctly evaluated? TP = trusted when good, TN = untrusted when bad, FP = trusted when bad, FN = untrusted when good.

Figure 6: Results for when resource-rich node 2 is good and 6 is unstable.

		wsn3 T	wsn3 U	wsn4 T	wsn4 U	wsn5 T	wsn5 U
rr2	AG	0.98	0.02	0.98	0.02	0.99	0.01
	AB	0.0	0.0	0.0	0.0	0.0	0.0
rr6	AG	0.43	0.08	0.43	0.08	0.43	0.08
	AB	0.08	0.41	0.08	0.41	0.08	0.41

Table I: Error matrices showing the percentage of time resource-constrained nodes (wsn) considered resource-rich nodes (rr) as being trusted or not. T = trusted, U = untrusted, AG = actually good, AB = actually bad.

Unstable Behaviour

- One always good edge node (rr2)
- One unstable (rr6)

Conclusions

- Cannot perform deterministic proactive trust assessment in asynchronous or synchronous systems
- Probabilistic is the best that can be achieved

Limitations:

- Proactive assessment does not assess willingness to perform the actual task
- How often a challenge is performed impacts the accuracy

Acknowledgement

- This work was supported by the PETRAS National Centre of Excellence for IoT Systems Cybersecurity EPSRC Grant EP/S035362/1.
- https://petras-iot.org

- You can find out more about the project at:
  - https://petras-iot.org/project/evaluating-trustworthiness-of-edge-based-multi-tenanted-iot-devices-team
  - https://mbradbury.github.io/projects/project-6-TEAM

Thank you for listening!