# *Easy as ABC*: A Lightweight Centrality-Based Caching Strategy for Information-Centric IoT

**Jakob Pfender**, Alvin Valera, Winston K. G. Seah

26 September 2019

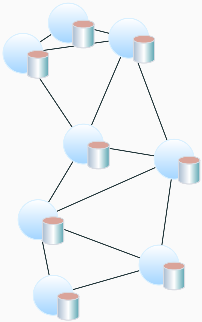
School of Engineering and Computer Science
Victoria University of Wellington

# Introduction & Motivation

# Caching in Information-Centric IoT

Benefits

- Fast information retrieval
- Reduced congestion
- Stability
- Decentralisation



# Caching in Information-Centric IoT

Benefits

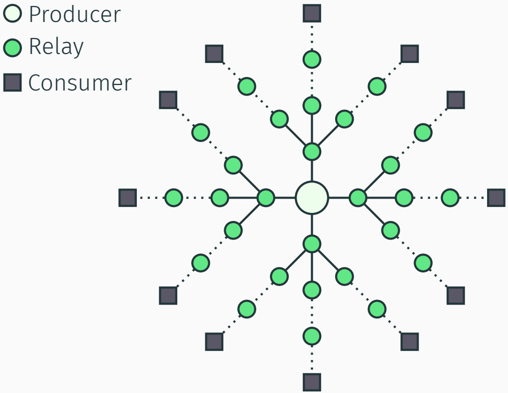
- Fast information retrieval
- Reduced congestion
- Stability
- Decentralisation

Challenges

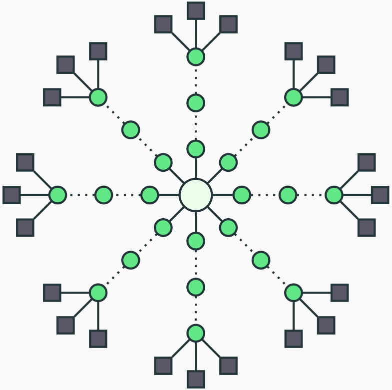
- **Limited memory**
- **Limited processing power**
- **Limited battery life**
- **Limited bandwidth**
- **Unreliable links**

# Where should content be cached?

# Multihop Topology Types in IoT




*Core topology*

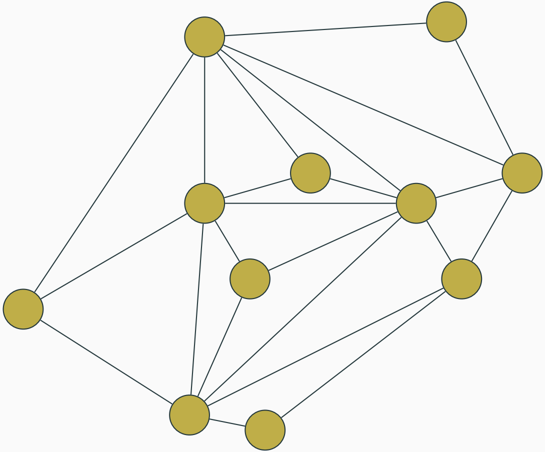


*Edge topology*

# Centrality-Based Caching Strategies

# Centrality-Based Caching Strategies

- Use caching node's **betweenness centrality** to decide where to cache content
- Betweenness centrality: The number of times a given node lies **on one of the paths** between all pairs of nodes in the network
- Centrality indicates node's **importance** in the network

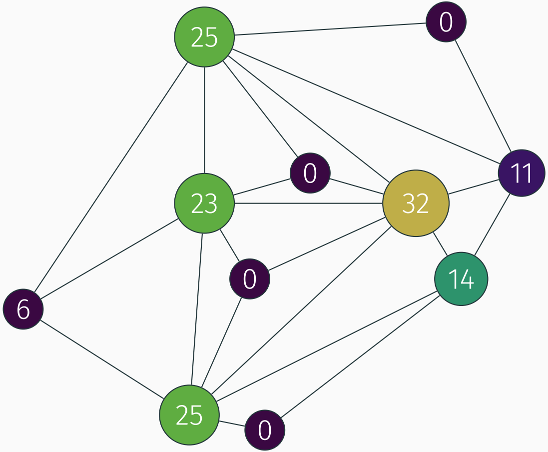


# Centrality-Based Caching Strategies

$$C_B(v) = \sum_{i \neq v \neq j \in V} \sigma'_{i,j}(v), \text{ where}$$

$$\sigma'_{i,j}(v) = \begin{cases} 1, & \text{if } v \text{ on path } (i,j) \\ 0, & \text{otherwise.} \end{cases}$$

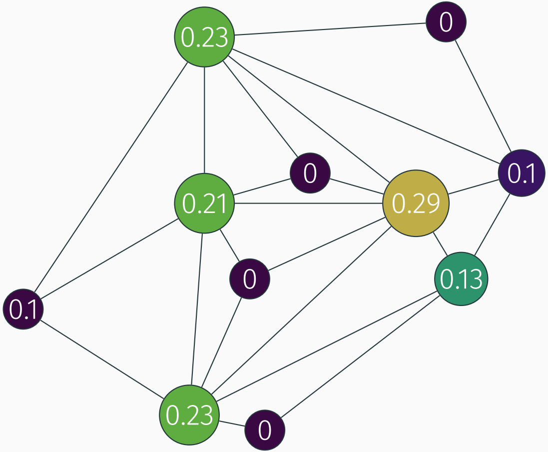
- Interest packets record **maximum** $C_B(v)$ among nodes they encounter
- Returning Data is cached at all nodes whose $C_B(v)$ is **equal to or greater** than maximum



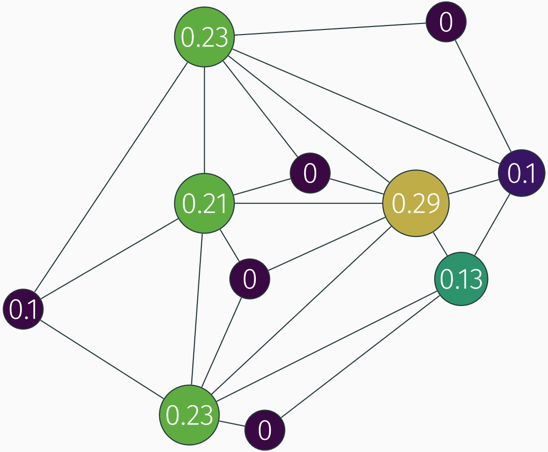
# *Betw* and *EgoBetw*

# *Betw*

- Each node's $C_B(v)$ is assigned **manually** (*a priori*) or through **exchange of neighbour information**
- For automatic assignment, every node needs **full information** about every other node
- **Full recalculation** required if topology changes

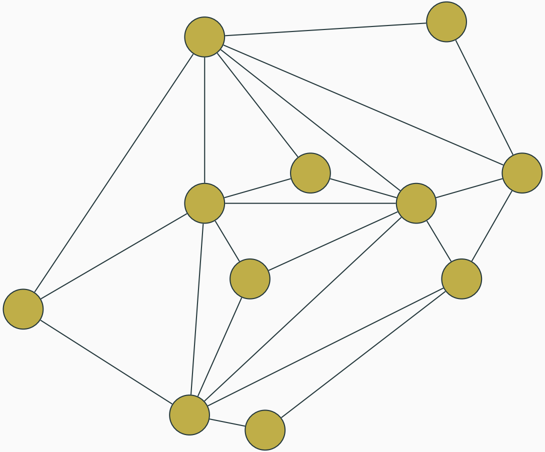


- Significant overhead:
  - Communication
  - Memory
  - Computation
- Complexity:
  - Messaging: $\mathcal{O}\left(n^2\right)$
  - Memory: $\mathcal{O}\left(n^2\right)$
  - Computational: $\mathcal{O}\left(n^2\right)$
- **Not feasible** for constrained networks

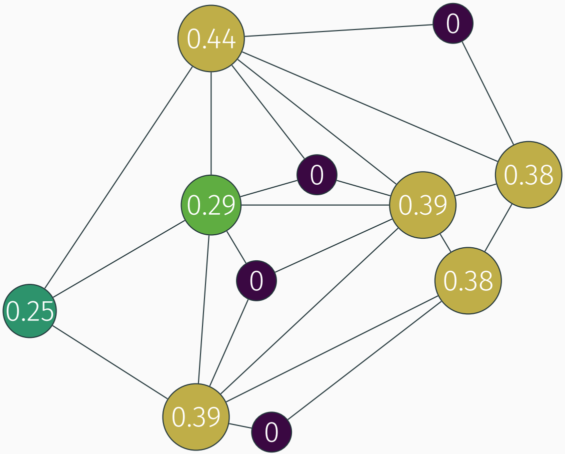


# *EgoBetw*

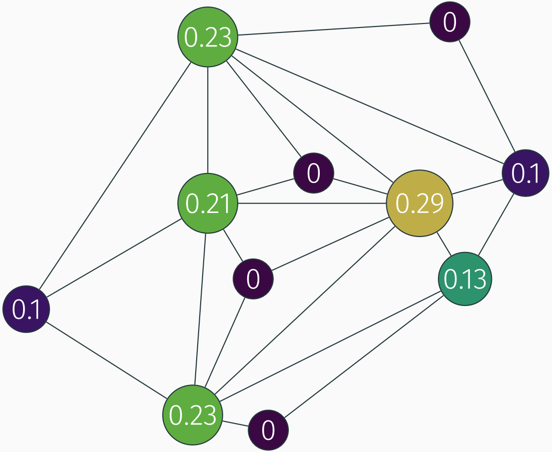
- Distributed solution
- **Ego network**: A node's one-hop neighbours and the links between them
- Centrality is calculated **only** for each node's ego network
- **Approximates** actual centrality

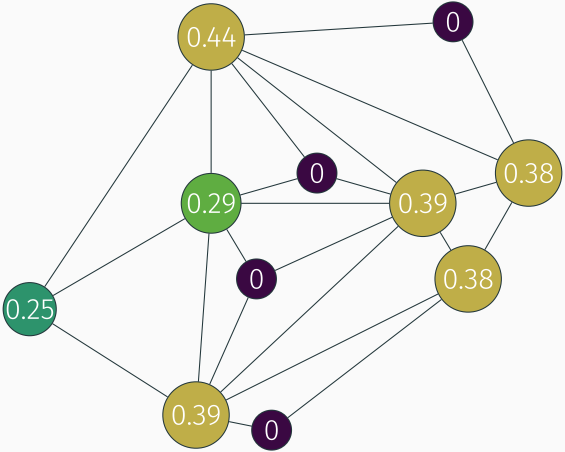


- Distributed solution
- **Ego network**: A node's one-hop neighbours and the links between them
- Centrality is calculated **only** for each node's ego network
- **Approximates** actual centrality



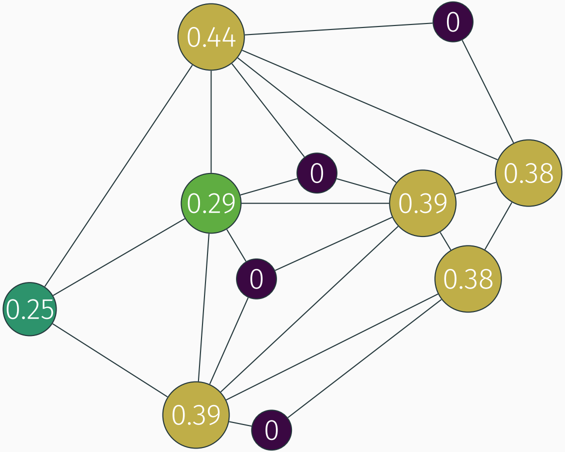
# *Betw* and *EgoBetw* — Comparison



# *Betw* and *EgoBetw* — Comparison

- **Loss of granularity** between absolute node centralities, but relative centralities **mostly** preserved
- **Reduced** complexity
  - Messaging: $\mathcal{O}(n)$
  - Memory: $\mathcal{O}(d^2)$ $(d \leq n-1)$
  - Computational: $\mathcal{O}(d^2)$
- Dynamic topologies slightly easier to manage



# *Betw* and *EgoBetw* — Summary

- *Betw* is **infeasible** for constrained hardware
- Existing research finds that *EgoBetw* delivers **satisfactory** approximation
- But its overhead is still **significant**!

# Our Goal

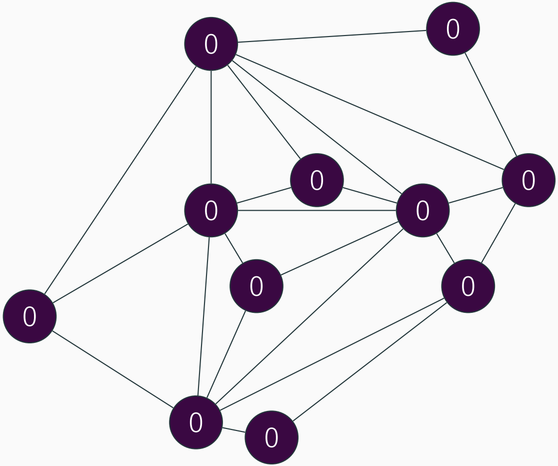
# Our Goal

Find a caching strategy that **approximates** the advantages of centrality-based caching while subject to the constraint that it **must be feasible** to implement and run on typical IoT hardware with **extremely limited memory and processing power**.

# Approximate Betweenness Centrality

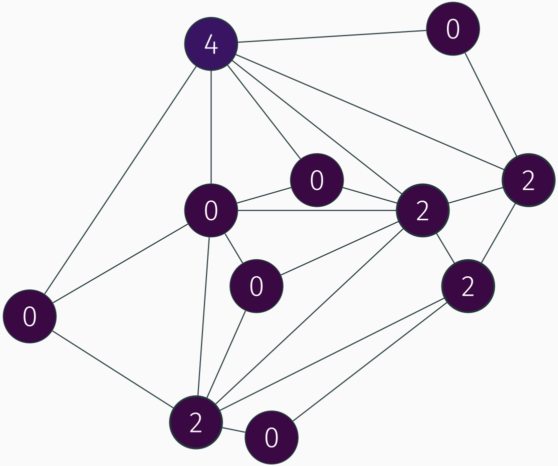
# Approximate Betweenness Centrality (ABC)

- Each node approximates its own centrality **during runtime**
- Using information **piggy-backed** onto normal ICN packets
- Interest packets are **extended** to carry UID of original requesting node



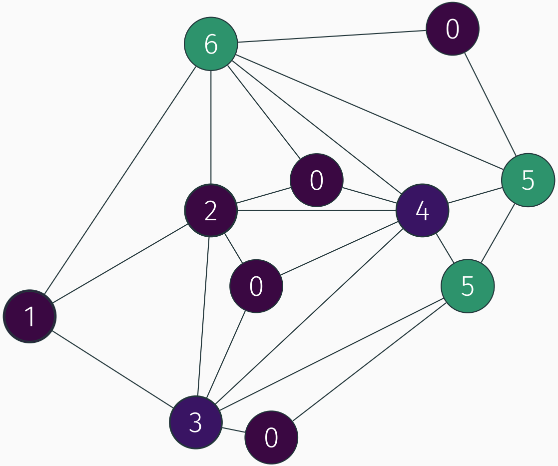
# Approximate Betweenness Centrality (ABC)

- Receiving an Interest means a node **knows** it is on the path between producer and consumer
- Each Interest from a new consumer or to a new producer **increases the node's knowledge**
- Over time, nodes can **approximate** their own centrality



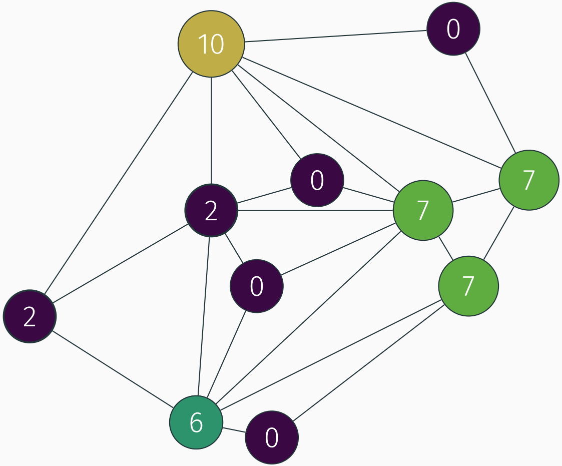
# Approximate Betweenness Centrality (ABC)

- Receiving an Interest means a node **knows** it is on the path between producer and consumer
- Each Interest from a new consumer or to a new producer **increases the node's knowledge**
- Over time, nodes can **approximate** their own centrality



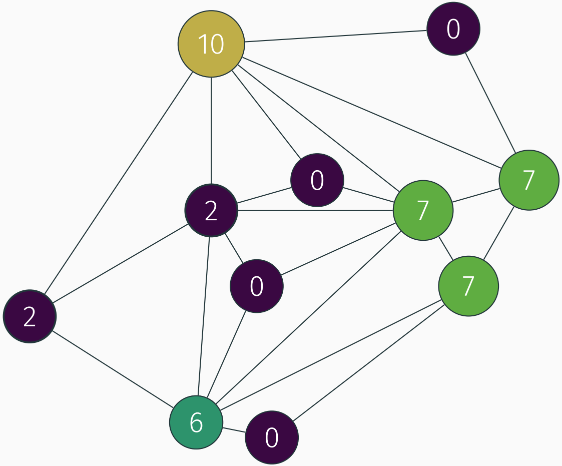
# Approximate Betweenness Centrality (ABC)

- Receiving an Interest means a node **knows** it is on the path between producer and consumer
- Each Interest from a new consumer or to a new producer **increases the node's knowledge**
- Over time, nodes can **approximate** their own centrality



# Approximate Betweenness Centrality (ABC)

- Introduces a **convergence time** ($\leq 60s$), but gets rid of *a priori* setup phase
- **Significantly reduced** complexity:
  - Messaging: $\mathcal{O}(1)$
  - Memory: $\mathcal{O}(p)$ $(p \leq n(n-1))$
  - Computational: $\mathcal{O}(1)$
- Can handle dynamic topologies by using **time-outs**
- Centrality values reflect **actual traffic patterns**



# ABC — Complexity Comparison

| Strategy | Messaging overhead | Memory overhead | Computational overhead |
|---|---|---|---|
| Betw | $\mathcal{O}(n^2)$ | $\mathcal{O}(n^2)$ | $\mathcal{O}(n^2)$ |
| EgoBetw | $\mathcal{O}(n)$ | $\mathcal{O}(d^2)$ | $\mathcal{O}(d^2)$ |
| ABC | $\mathcal{O}(1)$ | $\mathcal{O}(p)$ | $\mathcal{O}(1)$ |

$d$: Node degree ($d \leq n - 1$)

$p$: Number of paths ($p \leq n(n-1)$)

# Evaluation

# Evaluation — Experiment Setup

- All experiments conducted on **FIT IoT-LAB** open testbed using **M3 nodes**
  - STM32 (ARM Cortex M3), 512 kB ROM, 64 kB RAM, Atmel AT86RF231 2.4 GHz transceiver on IEEE 802.15.4
- Simple **RIOT-OS** application using **CCN-lite** as ICN implementation, modified to support the different caching strategies

# Evaluation — Experiment Setup

- All experiments conducted on **FIT IoT-LAB** open testbed using **M3 nodes**
  - STM32 (ARM Cortex M3), 512 kB ROM, 64 kB RAM, Atmel AT86RF231 2.4 GHz transceiver on IEEE 802.15.4
- Simple **RIOT-OS** application using **CCN-lite** as ICN implementation, modified to support the different caching strategies
- *Grenoble* site: $\sim$ 380 M3 nodes distributed evenly in a single building, **realistic indoor conditions** (multipath, reflection, absorption, interference)
- Choose 50 nodes randomly for each experiment
- Nodes can cache **up to 5 objects**

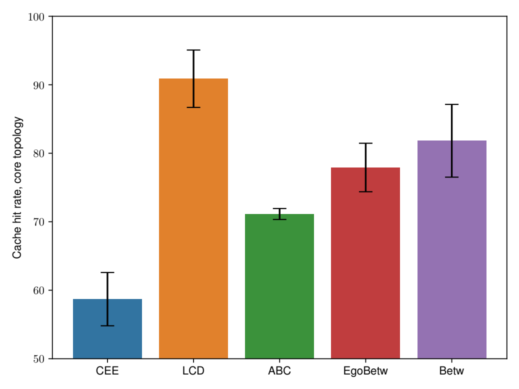
# Evaluation — Experiment Description

- Build **core** and **edge** topologies using routing algorithms and FIB assignment
- All nodes act as **producers, consumers, and relays**
- **Multihop** setup, average path length 3–4 hops

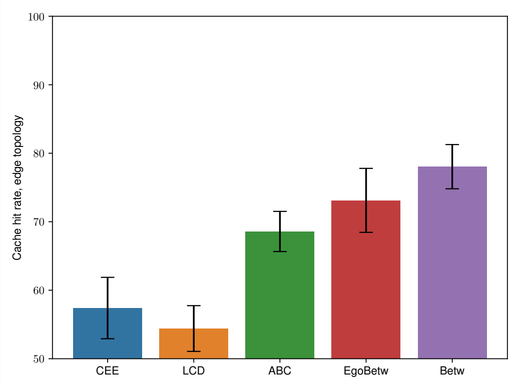
# Evaluation — Experiment Description

- Build **core** and **edge** topologies using routing algorithms and FIB assignment
- All nodes act as **producers, consumers, and relays**
- **Multihop** setup, average path length 3–4 hops
- Every node **periodically** requests content from producers following **uniform distribution**
- Content is cached according to **selected caching strategy**
- Strategies evaluated: Cache Everything Everywhere (CEE), Leave Copy Down (LCD), *Betw*, *EgoBetw*, ABC

# Evaluation — Cache Hit Rate

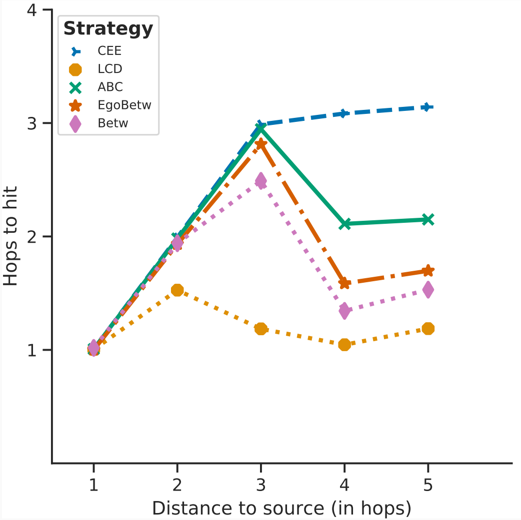



Core topology

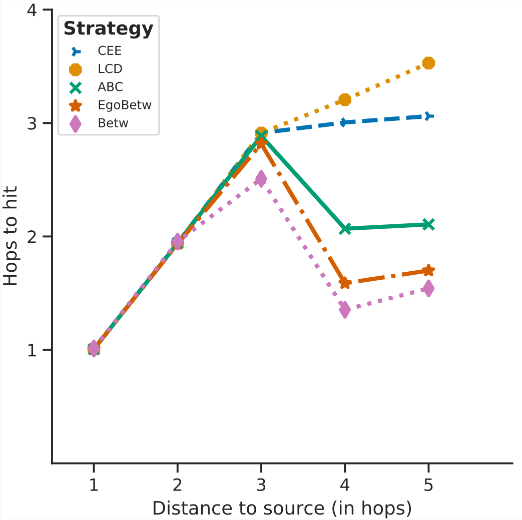



Edge topology

# Evaluation — Hop Count Reduction

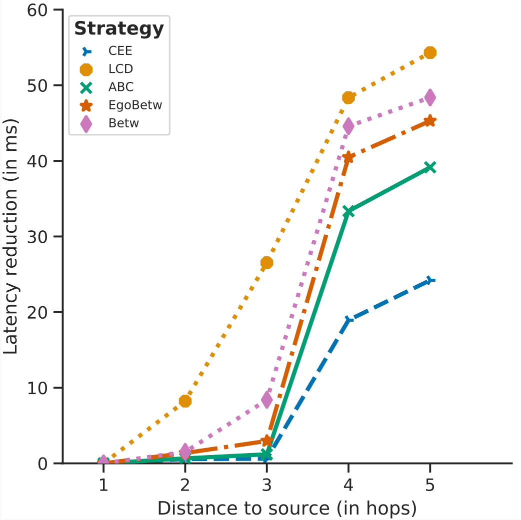



Core topology

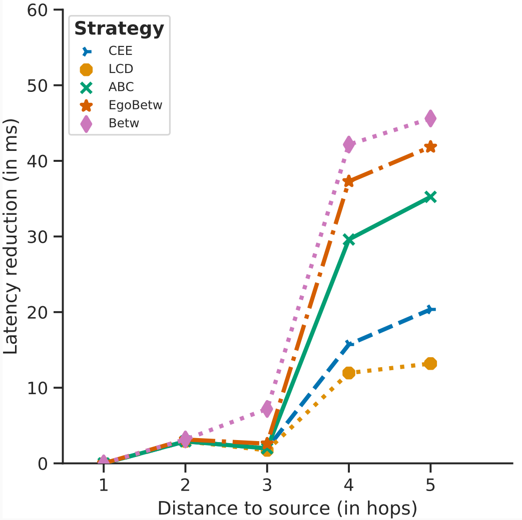



Edge topology

# Evaluation — Latency Reduction




Core topology




Edge topology

# Conclusions

# Conclusions

- Centrality strategies offer **significant benefits** for content delivery latency **regardless of network topology**

# Conclusions

- Centrality strategies offer **significant benefits** for content delivery latency **regardless of network topology**
- If the topology type is **known & static**, other strategies may be optimal

# Conclusions

- Centrality strategies offer **significant benefits** for content delivery latency **regardless of network topology**
- If the topology type is **known & static**, other strategies may be optimal
- Centrality strategies are a strong option if topology is **unknown or mutable**

# Conclusions

- Centrality strategies offer **significant benefits** for content delivery latency **regardless of network topology**
- If the topology type is **known & static**, other strategies may be optimal
- Centrality strategies are a strong option if topology is **unknown or mutable**
- ABC **not expected to outperform** existing centrality strategies because it relies on **less accurate** information

# Conclusions

- Centrality strategies offer **significant benefits** for content delivery latency **regardless of network topology**
- If the topology type is **known & static**, other strategies may be optimal
- Centrality strategies are a strong option if topology is **unknown or mutable**
- ABC **not expected to outperform** existing centrality strategies because it relies on **less accurate** information
- However, its complexity is **significantly lower** while results remain **acceptable**

# Conclusions

- Centrality strategies offer **significant benefits** for content delivery latency **regardless of network topology**
- If the topology type is **known & static**, other strategies may be optimal
- Centrality strategies are a strong option if topology is **unknown or mutable**
- ABC **not expected to outperform** existing centrality strategies because it relies on **less accurate** information
- However, its complexity is **significantly lower** while results remain **acceptable**
- It is **viable to implement** on constrained devices and is **consistent across topologies**

# Thank you!

Contact: jakob.pfender@ecs.vuw.ac.nz

# Classes of constrained devices

| Class | ROM | RAM |
|---|---|---|
| 0 | $<<$ 100 KiB | $<<$ 10 KiB |
| 1 | $\sim$ 100 KiB | $\sim$ 10 KiB |
| 2 | $\sim$ 250 KiB | $\sim$ 50 KiB |

(*Bormann* et al: *Terminology for Constrained-Node Networks, 2014*)

- RIOT 4.4 kB
- CCN-lite: 8.7 kB

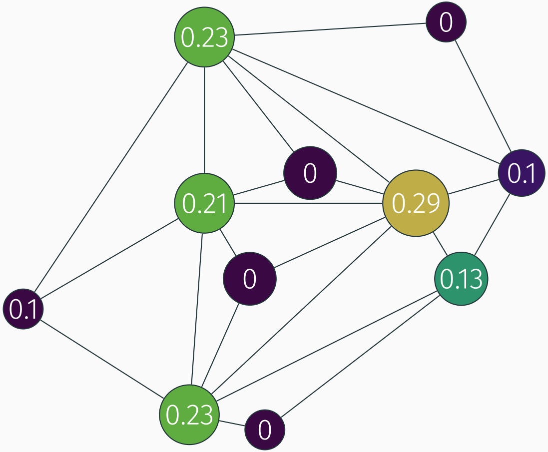
# Issues & Caveats

# ABC — Issues & Caveats

- Reliance on Interests that **clearly identify** producer and consumer
  - **Strong assumption**: Singular source for each prefix
  - **Break** in ICN principles: Carrying consumer information in Interest packets
- Single source assumption not generally true, but **not unrealistic** (uniquely identified sensor, rooms, etc.)
  - Can treat groups of nodes as **one producer** for path counting purposes
- Break with ICN principles unproblematic if application is siloed, but may **break interoperability** and **cannot provide anonymity**

# ABC — Issues & Caveats

- More central nodes are put under **more strain**
  - Exacerbated by already having to route more traffic due to position in topology
  - May lead to failure of the **most important** nodes in the network, thus defeating purpose



# ABC — Issues & Caveats

- More central nodes are put under **more strain**
  - Exacerbated by already having to route more traffic due to position in topology
  - May lead to failure of the **most important** nodes in the network, thus defeating purpose
- Potential solution: **Off-path caching**
  - Offload content to less strained neighbours
  - Can make use of centrality