Quality of Service for ICN in the IoT

draft-gundogan-icnrg-iotqos-01
ICNRG Interim, Macau

Cenk Gündoğan[1] Thomas Schmidt[1] Matthias Wählisch[2]

Michael Frey[3] Felix Shzu-Juraschek[3] Jakob Pfender[4]

[1]HAW Hamburg

[2]Freie Universität Berlin

[3]Safety IO

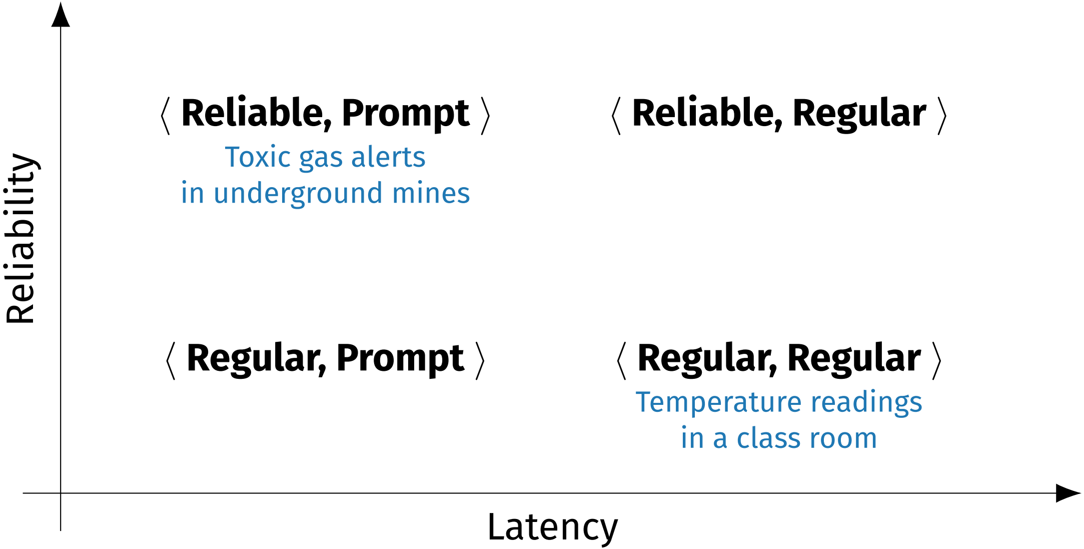
[4]VUW

September 27, 2019

Draft Positioning

- `draft-oran-icnrg-qosarch-01`
  - Outlines available resources and major differences for QoS in ICN vs IP
  - A strawman set of principles to guide QoS architecture for ICN
- `draft-moiseenko-icnrg-flowclass-04`
  - Proposes two methods for flow classification based on names
  - Uses indicators (additional TLV / name components) to map prefix to class
- `draft-oran-icnrg-flowbalance-01`
  - Maintain flow balance by accommodating wide dynamic range in Data MTU
  - Requester signals *expected Data size* in Interest message
- `draft-anilj-icnrg-dnc-qos-icn-01`
  - Uses name components to indicate routable part of name
  - Consumer adds QoS markers to non-routable part
  - Prefix matching of *PIT, CS, FIB* is adjusted accordingly
- `draft-gundogan-icnrg-iotqos-01`
  - Uses longest prefix match against preconfigured list for flow classification
  - **Focus:** Balance resources (link-layer buffer, CS, PIT) using correlations

Quality Dimensions



Reliability

⟨ Reliable, Prompt ⟩

Toxic gas alerts
in underground mines

⟨ Reliable, Regular ⟩

⟨ Regular, Prompt ⟩

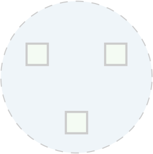
⟨ Regular, Regular ⟩

Temperature readings
in a class room

Latency

Resource Management Rules

1. Isolated Decisions

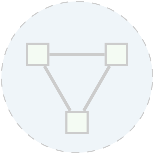


Forwarding Queue
Delay *regular* traffic

Pending Interest Table
Evict *regular* for *prompt*

Content Store
Evict *regular* for *reliable*

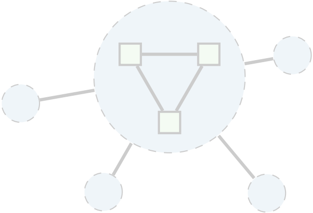
2. Resource Correlations



CS—PIT Correlation
Prompt Data meets no PI
⇒ cached with priority

CS—Forward. Correlation
Prompt Data dropped
⇒ cached with priority

3. Distributed Coordination

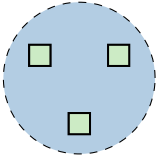


PIT Coherence
Same config. at all nodes
⇒ *Regular* < *Reliable* < *Prompt*

CS Efficiency
Same config. at all nodes
⇒ *Regular* < *Prompt* < *Reliable*

Resource Management Rules

1. Isolated Decisions

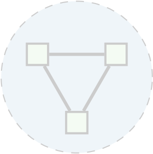


Forwarding Queue
Delay *regular* traffic

Pending Interest Table
Evict *regular* for *prompt*

Content Store
Evict *regular* for *reliable*

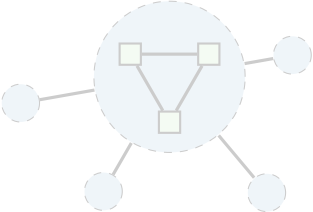
2. Resource Correlations



CS—PIT Correlation
Prompt Data meets no PI
⇒ cached with priority

CS—Forward. Correlation
Prompt Data dropped
⇒ cached with priority

3. Distributed Coordination

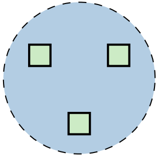


PIT Coherence
Same config. at all nodes
⇒ *Regular* < *Reliable* < *Prompt*

CS Efficiency
Same config. at all nodes
⇒ *Regular* < *Prompt* < *Reliable*

Resource Management Rules

1. Isolated Decisions

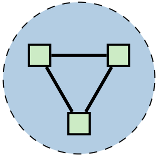


Forwarding Queue
Delay *regular* traffic

Pending Interest Table
Evict *regular* for *prompt*

Content Store
Evict *regular* for *reliable*

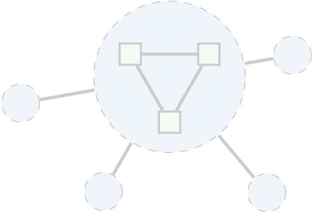
2. Resource Correlations



CS—PIT Correlation
Prompt Data meets no PI
⇒ cached with priority

CS—Forward. Correlation
Prompt Data dropped
⇒ cached with priority

3. Distributed Coordination

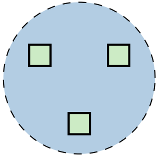


PIT Coherence
Same config. at all nodes
⇒ *Regular* < *Reliable* < *Prompt*

CS Efficiency
Same config. at all nodes
⇒ *Regular* < *Prompt* < *Reliable*

Resource Management Rules

1. Isolated Decisions

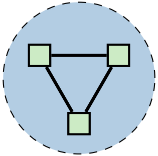


Forwarding Queue
Delay *regular* traffic

Pending Interest Table
Evict *regular* for *prompt*

Content Store
Evict *regular* for *reliable*

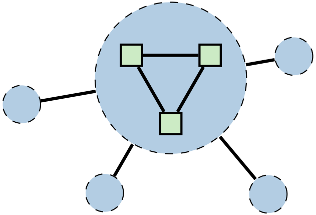
2. Resource Correlations



CS—PIT Correlation
Prompt Data meets no PI
⇒ cached with priority

CS—Forward. Correlation
Prompt Data dropped
⇒ cached with priority

3. Distributed Coordination



PIT Coherence
Same config. at all nodes
⇒ *Regular* < *Reliable* < *Prompt*

CS Efficiency
Same config. at all nodes
⇒ *Regular* < *Prompt* < *Reliable*

Next Steps

- Investigate more correlations between resources
- Examine risk of resource starvation
- Elaborate on distribution and maintenance of flow classes and service levels