Adaptive Data Replication in Real-Time Reliable Edge Computing for Internet of Things

Chao Wang, Christopher Gill, Chenyang Lu
What is being solved?

• For fault tolerance we need to replicate data.
• Replicating at the speed of arrival is inefficient.
• To come up with an adaptive data replication architecture for IoT edge computing that can meet applications’ latency and data-loss requirements with efficiency.
Challenges

• Sensing devices have limited storage capacity
• Limited network bandwidth of IoT gateways. Need to consider while deciding data replication.
• Applications have restrictions such as
  • Can tolerate only a certain number of data loses
  • End to end timing requirements
System Model and Analysis

![Diagram of edge computing for Internet of Things]

Fig. 1. Edge computing for Internet of Things.

Publish subscribe model
System Model and Analysis

Key terms and Notations
• Data topics
• $T_i$ - Minimum inter-publishing time for data topic $i$
• $N_i$ – Data elements that a publisher can keep for a data topic $i$
• $L_i$ – Maximum number of consecutive losses that a subscriber can accept for a data topic $i$.
• $D_i^P$ – latency requirement – soft end-to-end deadline.
• $D_i^R$ – relative replication deadline

Fig. 1. Edge computing for Internet of Things.

Publish subscribe model
System Model and Analysis

Observations that were used to come up with the categories:
- Data publishers have limited data storage for retransmission
- Data topics may have moderate or no loss-tolerance requirements – inference tasks where data loss can be compensated by estimation.
- Data topics may require zero loss but have no latency requirement - logging

<table>
<thead>
<tr>
<th>Category</th>
<th>( L_i )</th>
<th>( N_i )</th>
<th>( D_i^P ) (ms)</th>
<th>( T_i ) (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>( \infty )</td>
<td>50</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
<td>500</td>
<td>500</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>0</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>0</td>
<td>100</td>
<td>100</td>
</tr>
</tbody>
</table>

TABLE I
Example Data Topic Specification.
When do we actually need to replicate data?

• Let $x_i(t)$ be the largest number of consecutive uncovered data elements, the system meets fault tolerance if $x_i(t) \leq L_i$ at all times.

• For small $N_i$: $x_i(t)$ depends on:
  • $T_i$
  • Edge computing task time
  • Scheduling

• No replication needed if elements are processed before new elements are sent if $L_i \geq 1$.

• Regular data replication can be used based on some predetermined conditions.
When to do the replication?

• A deadline is decided
• 2 lemmas help us understand the constraints better and come up with an architecture:
  • For data topic $i$, to prevent more than $L_i$ consecutive data losses, $L_i$ and $N_i$ cannot be both zero.
  • For data topic $i$, set parameter $M_i \geq 1$ and let $y = L_i - M_i$. To prevent more than $L_i$ consecutive data losses, the replication deadline must satisfy the following bound:
    • $D_i \leq (N_i + y + 1)T_i - T_{\text{FO}} - \delta_{\text{PP}} - \delta_{\text{PrB}}$
      • $T_{\text{FO}}$: Fail over time
      • $\delta_{\text{PP}}$: Latency from publisher to Primary
      • $\delta_{\text{PrB}}$: Latency from Primary to Backup
  • Lemma 2 implies that a shorter interval between replications (a smaller $M_i$) can permit a longer replication deadline.
ARREC Architecture

Mark → Wait → Clear → Batch

arrivals of different data topics

global earliest deadline to start a replication

batch window

d_i: deadline to start replicating data arrived at t_i

Data publishers

The Primary
Edge computing engine
Replication handler
Recovery handler
Edge computing engine

Data subscribers

The Backup
Empirical Analysis

- Two $M_i$ configurations for AAREC – 1 and $L_i$: Two extremes
- Baseline:
  - Retransmission-only: No replication. Used to understand overhead of replication.
  - Periodic: 50ms (shortest in topic categories) and 25ms
Empirical Analysis