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

+Chao Wang, *Christopher Gill, *Chenyang Lu

+ Department of Computer Science and Information Engineering, National Taiwan Normal University
* Department of Computer Science and Engineering, Washington University in St. Louis, USA
Industrial Applications: Deadlines

• Reliable and timely computing at a resource-constraint network (an IoT gateway)

  • Single point of failure
  • Traffic congestion and delay
  • Limited network bandwidth

Example: structural health inference and control
Models

• Application (Data subscriber)
  • Sensors generate data every $T_i$
  • Data received by IoT Gateway/Edge and a task computes result at some point
  • Result sent to application which has a deadline and loss tolerance

• System
  • Sensors generate data periodically and can buffer limited data $N_i$ => do not fail
  • IoT Gateway/Edge node receive data and compute a task => infinite buffer
  • IoT Gateway/Edge node: primary and backup => primary can fail

• Coverage
  • Data is buffered at the sensor
  • Backup has a copy
  • Result has already been completed and delivered
Specific IoT gateway requirements

• Quantitative requirements
  • Data subscriber imposes an end-to-end soft deadline for data result $D_i$
  • Data subscriber cannot accept more than $L_i$ consecutive losses for data topic $i$

• Qualitative requirements
  • The gateway should not consume too much local network bandwidth
Tradeoff

Replication can reduce data loss

Increases amount of bandwidth consumption; primary sends data to backup gateway
System model

• Publish-subscribe data model
  • with a minimum inter-publishing time for each topic $T_i$

• Each embedded sensing device can only keep $N_i$ latest data elements

• Primary-backup fault tolerance
  • crash failures (fail-stop)
  • data replication to backup
  • fail over
Example Data Topic Parameters

<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.

- e.g. logging – no loss, no time constraint
- Interesting: no latency requirement under EDF means greater data loss!
Key idea for adaptive data replication

• In the IoT gateway, once data is processed/delivered, it is irrelevant

• Therefore, for each data, we may postpone replication activities to reduce the need of actually performing the replication

• Must replicate in batch window to meet loss and timing requirements
• Once started, perform a batch of pending replications for efficiency
• Once batch window is filled, any new replications occur in next batch window
Adaptive data replication architecture

- Edge computing engine schedules both computing tasks and replication tasks using the EDF policy.

- Replication handler decides the intended rate of replication $M_i$ (parameter) from $M_i$ to compute replication deadline.

*Lower intended rate -> tighter replication deadline*

(see the paper for the analysis)
Empirical performance: efficiency in network bandwidth usage

A higher intended replication rate is preferred, because it permits a longer replication deadline, which in turn would allow the system to skip many more replication activities.

88% saving in bandwidth.
(Payload size = 512 bytes)
Summary

• Positives
  • reliability, timeliness, and resource constraints
  • key idea: adaptive data replication
  • failure rate agnostic?

• Negatives
  • weak failure model
  • is host-host bandwidth (gateway-gateway) really an issue?