Service Level Guarantee for Mobile Application Offloading in Presence of Wireless Channel Errors

ARANI BHATTACHARYA, STONY BROOK UNIVERSITY & SUNY KOREA
ANSUMAN BANERJEE, INDIAN STATISTICAL INSTITUTE
PRADIPTA DE, GEORGIA SOUTHERN UNIVERSITY
Why Computation Offloading?

- It can provide **fast response time** to user!

Client

Compute-intensive Applications

**Slow Processor**, Low memory, Small battery ...

[Image of a smartphone with icons and text boxes indicating compute-intensive applications and slow processor issues]
How Offloading Works?

- Mobiles use wireless network to communicate with cloud
Some concerns?

- Battery consumption
- Parallel application
- Different mobile system
- Real-time Constraints

Low latency

Game interaction control

Video stream
Some concerns?

- Battery consumption
- Parallel application
- Different mobile system
- Real-time constraints

Server

Low latency, Channel Error

Client

Game interaction control

Video stream

Cloud Gaming
Some concerns?

- Offloading Gain

<table>
<thead>
<tr>
<th>Finish Time (s)</th>
<th>Channel Error (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Estimated</td>
<td>Actual</td>
</tr>
<tr>
<td>Oracle</td>
<td>Local</td>
</tr>
</tbody>
</table>

Low latency, Channel Error

Server

Game interaction control

Video stream

6
Our contribution

- Design an algorithm
  ◦ Ensuring service level guarantee (low failure rate)
  ◦ Executing faster than local execution
  ◦ Dealing with unpredictable channel errors properly
Our Simple Example

- Offload one frame of data; channel error = 0; failure rate = 0.01.

1 attempt -> 0.2 (not satisfy)
2 attempts -> 0.04 (not satisfy)
3 attempts -> 0.008 (satisfies)
Problem Model

Application

Network

Task

Dependency

Data Packet

Channel With Errors

Delivered Frames

Frame With Error
Our Algorithm

1. Check channel condition
2. Calculate number of retransmission attempts
3. Faster than Local?
   - Yes: Offload
   - No: Execute Locally
Formulation: Objective

Minimize Expected Finish time

\[ \text{Min } E[T_m] \]
Formulation Constraint

Service Guarantee Constraint

\[ P(T_m > U_m) \leq \varepsilon \]

- Execution Finish Time
- Local Execution Time
- Failure rate
Formulation: Summary

Min $E[T_m],$

Subject to:

$P(T_m > U_m) \leq \varepsilon$
Reduction of Service Guarantee

Overall Failure = $\bigcup P(Individual\ Migration\ Failure)$

$\epsilon \geq \sum \alpha_i^s + \alpha_i^r$

Overall Failure rate  
Failure rate of sending one packet  
Failure rate of receiving one packet

We choose $\alpha_i^s + \alpha_i^r = \frac{\epsilon}{2i}$
Single Migration Constraint

\[ P(T_m > U_m) \leq \alpha^s \]

Assumptions:

1. Each frame takes equal time
2. One packet fragments into \( w \) frames
Single Migration Constraint

How many retransmission attempts satisfy constraint?

Migration: \( Q \sim \text{Binomial}(z, p) \geq w \)

- Random variable Denoting Number of transmissions
- Number of maximum Transmission attempts
- Probability of successful transmission
- Number of frames

How to calculate \( z \) given \( p \) and \( w \)?
Calculating transmission attempts

No direct formula to get exact value

Our solution: Use Hoeffding’s inequality to approximate $z$:

$$z \geq \frac{p(w-1)(4+\sqrt{2})-\ln(\alpha^s)}{2p^2}$$

Similarly, for receiving packet from cloud to mobile,

$$z \geq \frac{p(w-1)(4+\sqrt{2})-\ln(\alpha^r)}{2p^2}$$

Expected time maximized when $\alpha^s = \alpha^r$
Evaluation

The diagram shows the failure rate (%) for different values of $\tilde{p}$, which are $5\%$, $10\%$, $20\%$, $30\%$, $40\%$, and $50\%$. The failure rate is plotted against the number of iterations (ILP, 1, 5). The graph indicates that the failure rate decreases as the number of iterations increases, especially for lower values of $\tilde{p}$. This suggests that the system is more robust with higher iteration counts.
Finish Times

![Graph showing finish times for different failure rate bounds (ε) and p values.]
Summary

➢ Providing guarantee of service useful in offloading over uncertain network

➢ Single transfer of data is modeled as a Binomial process
  ➢ Number of transmission attempts obtained using Hoeffding’s inequality

➢ Our algorithm developed can provide stochastic guarantees

Questions?