Towards Mobile Opportunistic Computing

Abderrahmen Mtibaa, Khaled A. Harras, Karim Habak, Mostafa Ammar, Ellen W. Zegura
Motivation

• Mobile devices are everywhere!
• Mobile application explosions
  ○ Many “heavy” apps: Face recognition, body language interpretation, NLP, speech and object recognition...etc.
  ○ *Exceed the limits* of stand-alone mobile devices
• Solution has been: Computation Offloading
• Offloading to *cloud*
  ○ MAUI [MobiSys’10], CloneCloud [EuroSys’11], COSMOS [MobiHoc’14], etc.
• Recently...
  ○ Bringing computation closer to mobile (e.g. Cloudlets)
  ○ Offloading to *neighboring devices* (e.g. Serendipity [MobiHoc’12], FemtoCloud[CLOUD’15], and MDCs[CloudCom’13])
**Motivation...**

**Goals:**

Rethink the perspective of mobile computation offloading.

Propose a dynamic and adaptive peer-to-peer offloading architecture.

Leverage ALL computation (a-d) & communication (BT, WiFi, etc.) opportunities.
Outline

- Motivation
- Opportunistic Computing Challenges
- Peer-to-Peer Opp. Computing Architecture
- Implementation Prototype
- Experiment & Preliminary Results
- Conclusion & Future Work
Opportunistic Computing Challenges

- **Intermittent connectivity**
  - Connectivity is often variable and unknown
  - Difficult to guarantee delivery of task or results
  - Providing protection against future network disruptions.

- **Partitioning and remote execution**
  - Unknown connectivity and device capabilities to apps.
  - Efficient partitioning between the devices

- **Heterogeneous computation and communication resources**
  - Need for a common architecture for computation offloading
Peer-to-Peer Opportunistic Computing Architecture

- Offloading Manager: heart of the architecture, scan, detect
- Computation Opportunities: the offloading manager selects based on tasks and devices
- Profiler: For each task T, nb of subtasks, complexity, input, output, etc.
- Databases: store & access social, comp. and comm data
- Internal computing resources: run a task scheduled for local execution
- Task Profiler
- App.
- Offloading Manager
- Discover & Negotiate use of computing resources
- Incentives Manager
- MDC, Cloudlet, Cloud, Local
- Resource Allocation Manager
- Routing & Replication
- Task Scheduler
- Forwarding Manager
- Interface Scheduler, QoS & Guarantee, Fault Tolerance
- Computing Resources: processor, energy, storage, etc.
- Communication Interfaces: WiFi Direct, Bluetooth, WiFi, 3G/4G
- Forwarding Manager
- Interface Scheduler
- QoS & Guarantee
- Fault Tolerance
- Internal computing resources: run a task scheduled for local execution
- Databases: store & access social, comp. and comm data
- Offloading Manager: heart of the architecture, scan, detect
- Computation Opportunities: the offloading manager selects based on tasks and devices
- Profiler: For each task T, nb of subtasks, complexity, input, output, etc.
- Forwarding Manager
- Interface Scheduler, QoS & Guarantee, Fault Tolerance
- Computing Resources: processor, energy, storage, etc.
- Communication Interfaces: WiFi Direct, Bluetooth, WiFi, 3G/4G
- Task Scheduler
Proof-Of-Concept Prototype

- We build an *android application*
  - same app for offloader and offloadee
  - Collaborative mode (enable/disable at any time)
  - User authorizes its communication medium that can be used by the app.

- Our app. implements the *task scheduler*
  - determine technology and offloadee to use
  - maximize the *utility function* (time, energy, etc.)

- App. specifies the maximum number of threads to run on the offloadee device
Experimental Results

- Dummy tasks running at the offloader
  - Execution either locally or offloaded to 1+ offloadee

- Each task consists of
  - Computation requirements (measured in FLOPs)
  - Input and output data (measured in Bytes).

- Evaluation scenario
  - 3 devices: 1 GS5, 1 N7, and 1 N10
  - Offloadee has 3 tasks $t_1$, $t_2$, $t_3$

- We measure required time to:
  - run tasks locally
  - run tasks remotely using our app.
Conclusion & Future Work

- We propose a novel peer-to-peer architecture for mobile opportunistic offloading
  - Routing, scheduling, discovering, securing, etc.
- We have also implemented a proof-of-concept prototype
  - Android application implementing the task scheduling
  - Achieve $3 \times$ speedup in execution time using only 3 mobile devices.
- Our future work consists of:
  - Investigate different routing strategies,
  - Propose objective functions, incentive systems, and QoS & guarantee models.
Thank You!

amtibaa@cmu.edu