Optimizing Operation of a Hierarchical Campus-wide Mobile Grid for Intermittent Wireless Connectivity

Konstantinos Katsaros and George C. Polyzos

Mobile Multimedia Laboratory
Department of Computer Science
Athens University of Economics and Business

Outline of Talk

- Motivation
- Related Work
- Proposed Architecture
- Evaluation Framework
- Results
- Conclusions & Future Work

Motivation

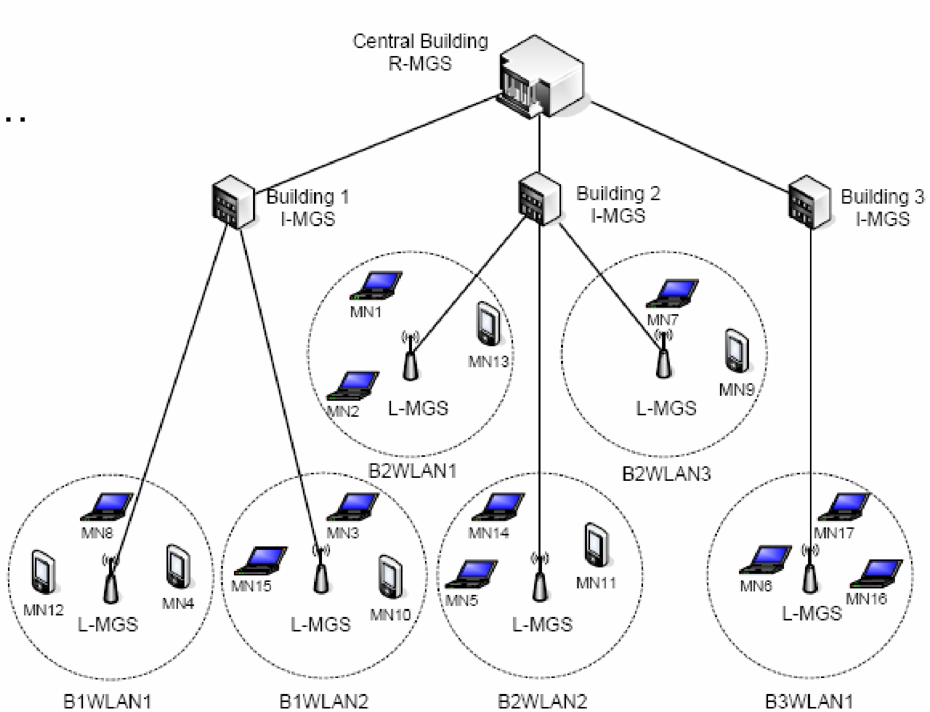
- Grid computing *paradigm*
  - "…coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations."
  - Large-scale distributed system designed to aggregate resources from multiple sites.
  - Strong interest of the scientific community
- Mobile computing *paradigm*
  - Enormous number of mobile computing devices
  - Resource limitations (e.g. CPU, storage, power) but…
  - … increasingly becoming more powerful!
- Merging the two *paradigms*
  - What should be the nature of such merge?

Related Work

- ***Mobile Devices as Resource Consumers***
  - Access to fixed Grid infrastructure
  - *Target:* Provide the missing resources
  - Problems due to *mobility*, *wireless* interface, *heterogeneity*
  - Proposed solution: *proxies/mediators a*ct on behalf of the *Mobile Node* (MN)
- ***Mobile Devices as Resource Providers***
  - Enormous number of increasingly powerful mobile devices
  - *Target:* Aggregate all these scattered resources
  - Same problems remain
  - ***Mobile Grids On-Site (***Infrastructure mode)
    - Aggregation of resources residing in a *Service Area* (SA), e.g. WLAN, cell.
    - Central co-ordination: service discovery, job splitting, task assignment, monitoring etc.
  - ***Mobile Ad-Hoc Grids***
    - Completely distributed
    - Further problems: No central co-ordination, Network partitioning, Multi-hop routing
    - Proposed solution: *virtual backbone i.e.* more powerful MNs act as coordinators

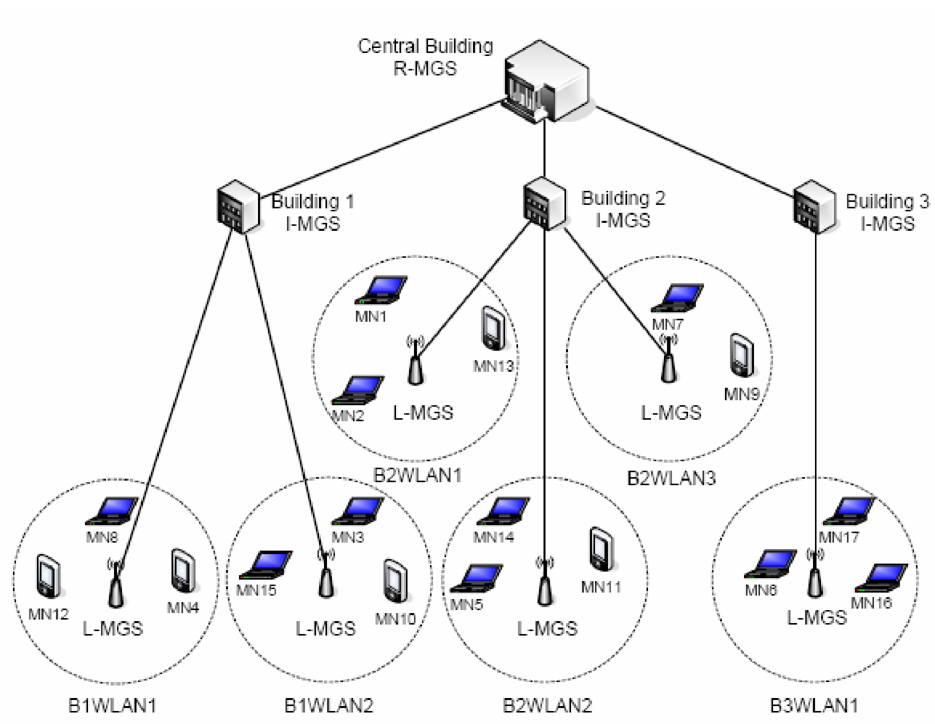
Proposed Architecture (1/2)

- *Mobile Grid On-Site* approach
- *Mobile Grid Schedulers* (MGSs)
  - Receive a job from the upper level …
  - …decompose it into tasks and …
  - …assign tasks to lower levels
  - Receive & combine the results
  - Propagate information on available resources upwards
- *Root*-MGS (R-MGS)
  - Can receive job submissions from outside the campus
- *Intermediate*-MGS (I-MGS)
- *Local*-MGS (L-MGS)
  - Serving a WLAN
  - Assigns tasks to MNs
  - Can also receive job submissions from MNs



Proposed Architecture (2/2)

- Jobs submissions may be propagated upwards
- Results may be returned from a different point of attachment
- *Hierarchical* structure
  - *Divide-and-conquer* approach
  - *Levels of abstraction*
  - Load balancing
- *Campus-wide*
  - Large number of MNs
    - E.g. ~6200 distinct MAC addresses recorded at Dartmouth campus
  - Central administration



Incentives

- *Why should Mobile Nodes share their resources?*

  Reciprocity
  - A mobile node is allowed to submit a job only if it offers its own resources a well
  - MNs take advantage of the aggregated resources
- *Why not each MN compute its own jobs?*
  - A whole job may require resources not available in a single MN
  - In a certain period of time:
    - Offered resources << Required resources
    - Small amounts of resources offered by several MNs
    - Taking advantage of the parallel character of task execution
- Fairness issues (e.g. *free-riding)* demand for an *accounting* mechanism

Evaluation Framework

- *Divisible Load* applications: the load of computation can be divided in several independent parts
- Three-step process: $T_{TOTAL} = T_{IN} + T_{EXEC} + T_{OUT}$
- *Communication to Computation Ratio (CCR):*

$$CCR = \frac{CommunicationCost}{ComputationCost}$$

- Performance depends on the actual *Response Time* (RT)
- Intermittent connectivity imposes *delays* on T_{IN} *and* T_{OUT}

$$Overhead = \frac{RT - T_{TOTAL}}{T_{TOTAL}} * 100\%$$

  - On the utilized trace set (*Kotz et al.*):
    - Mean connection time: 16.6 minutes,
    - 71% less than 1 hour,
    - 27% less than 1 minute!
- However, there is a distinction between disconnection and failure.

Task Replication

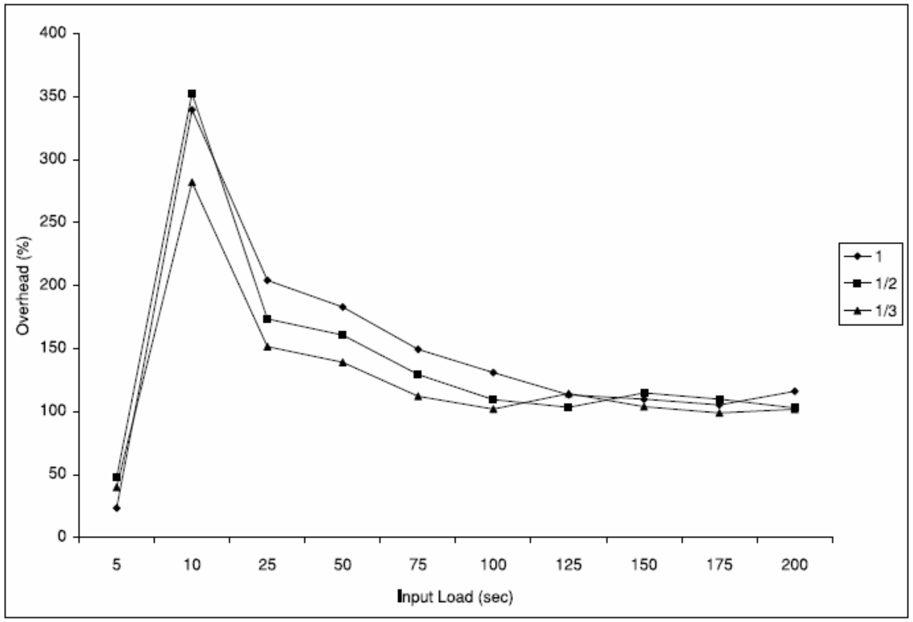
- Assigning the same task to more than one MNs in the same WLAN
  - Not all MNs present the same networking behavior
  - Some will eventually return the results earlier than the others
- Resource waste
  - Resources on MNs performing worse
- *Tradeoff*:
  - The greater the extend of task replication the larger the size of the task
  - Probability of disconnection increases

Traces

- WLAN mobility traces from the Dartmouth University campus
- In the form of : (`MN,AP,timestamp`)
  - Special AP name for disconnection: “OFF”
- Collected from April 2001 to March 2003
- Subset used due to “holes”:
  - Duration: 01January 2002 – March 2003
  - 5982 distinct MAC addresses
  - 566 APs
  - 166 buildings
- Almost 1000 *testing environments i.e.* <*`time, AP`*>
  - Uniformly distributed across the trace set

Results: *Delay Overhead (1/2)*

- Low overhead for very low input loads
  - Process completes before disconnection
- Dramatic increase
  - Disconnections during data transfer
  - Low T_{TOTAL}
- Overhead decreases for higher input loads
  - Computation step compensates for disconnection

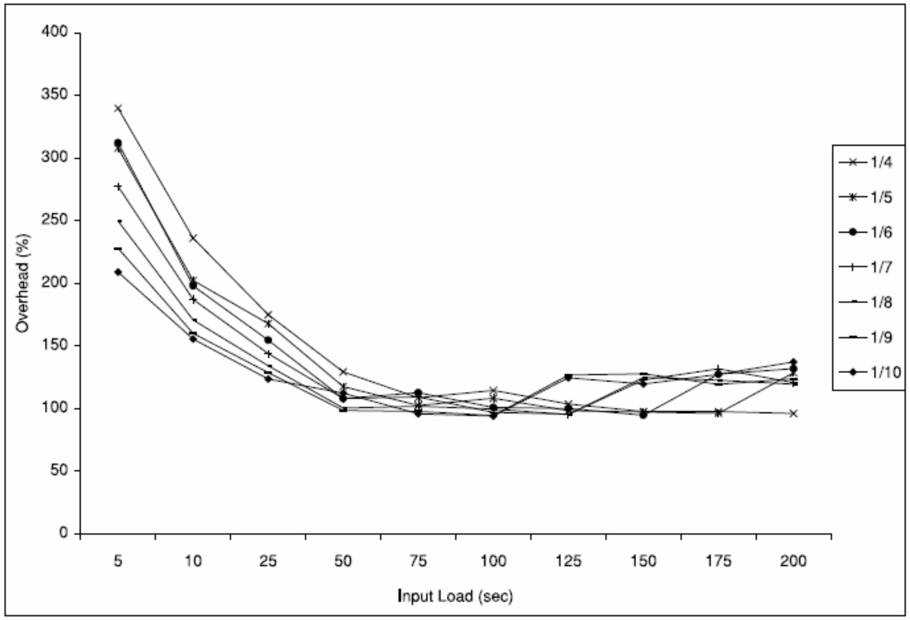


Overhead for high CCR values

- Load measured in *time* units (seconds)
  - No information on the actual throughput
  - More general framework
- $T_{IN} = T_{OUT}$

Results: *Delay Overhead (2/2)*

- For low input load: 210% - 340% overhead
  - Low T_{TOTAL}
- Low CCR values result in higher T_{TOTAL}
  - Probability of disconnection during transmission of the results increases
- For low input volumes, lower CCR values result in lower overhead
  - Computation during disconnection
- As the input volume increases, higher CCR values become preferable
  - Lower T_{TOTAL} and lower probability of disconnection



Overhead for low CCR values

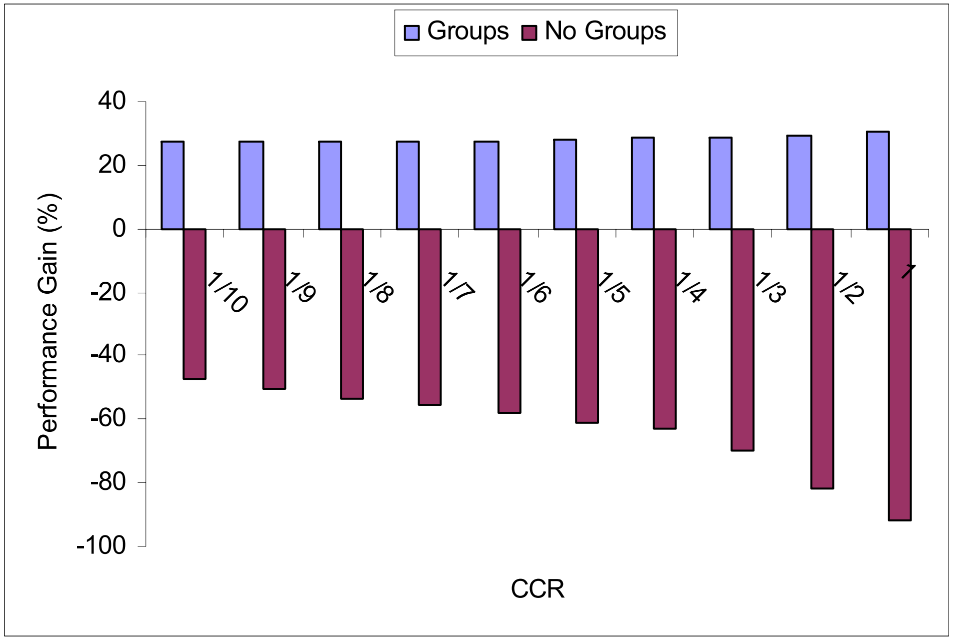
Results: *Task Replication (1/3)*

- Extend of replication subject to number of MNs co-residing in a WLAN
  - On average: 4 MNs, in the utilized set of traces
  - Hence, task replication for 2 groups of MNs in a WLAN (*GROUPS case)*
- Performance compared to the “no replication” scenario (*NO GROUPS case*), in terms of:
  - Overhead (compared to the performance of a single MN…)
  - Percentage of success scenarios…

Results: *Task Replication (2/3)*

- Superior performance for the *GROUPS case*
  - *GROUPS* case leverages the parallel character of task execution (28% performance gain on average)
  - *NO GROUPS* case incurs an overhead of 57% on average
  - Worse for high CCR values due to limited compensation of disconnection periods with computation

- *Performance Gain: RT* vs. T_{TOTAL}
- For various T_{IN} values *(5-200 sec)*
- $T_{IN} = T_{OUT}$

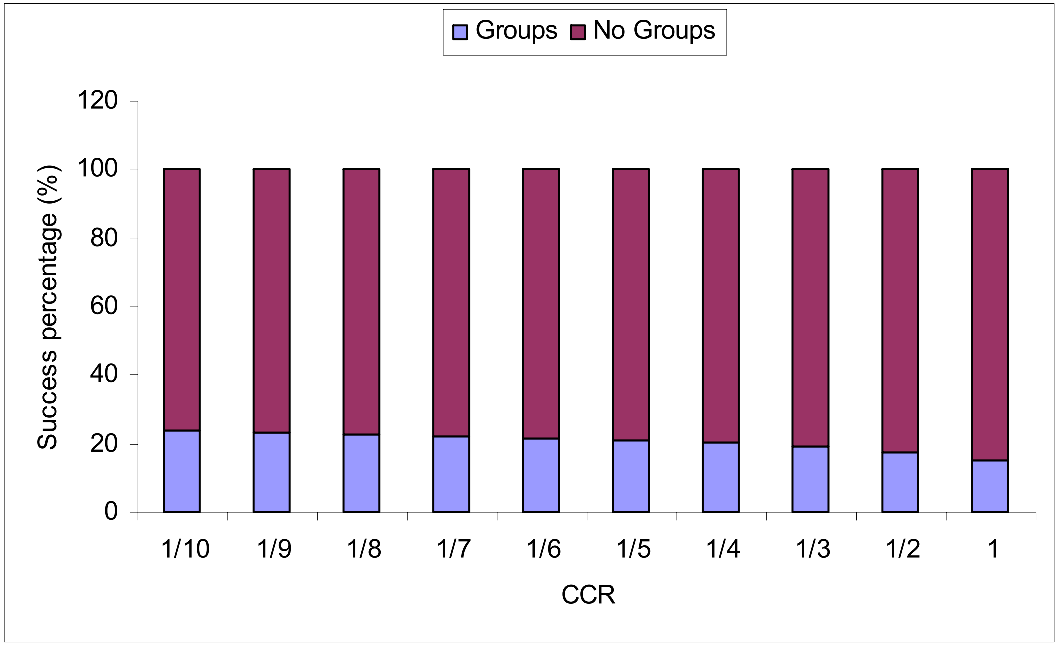


Performance gains by Task Replication scheme

but…

Results: *Task Replication (3/3)*

- In the vast majority of the testing environments the *RT* of the *NO GROUPS case* is lower.
  - Due to the increased (i.e. double) task size in the *GROUPS* case
  - As the total load increases (lower CCR values), a slight increase is noticed in the percentage of preferred *GROUPS case* testing environments



Percentage of success scenarios

Conclusions & Future Work

- A hierarchical campus-wide networking environment seems a realistic context for Mobile Grid
- Processing step can *hide* mobility problems
- Task replication: promising technique for heavy load/instable application/networking environments.

- Detailed *incentives* scheme and *accounting* mechanism
- *Load balancing* throughout the MGS hierarchy
- *Modeling* of MN's networking behavior
  - Target: *Expected Response Time*
  - Valuable input for L-MGSs: MN selection, Job decomposition: task size

Thanks!

Questions ?

Konstantinos Katsaros

Mobile Multimedia Laboratory
Department of Computer Science
Athens University of Economics and Business

ntinos@aueb.gr

http://mm.aueb.gr/