Dynamic QoS-Aware Coalition Formation

Luís Nogueira, Luís Miguel Pinho

{luis,lpinho}@dei.isep.ipp.pt

IPP-Hurray! Group

School of Engineering, Polytechnic Institute of Porto

Porto, Portugal

Outline

- Problem statement
- Proposed approach
- QoS requirements specification
- Coalition formation
- Proposals evaluation
- Proposals formulation
- Conclusions

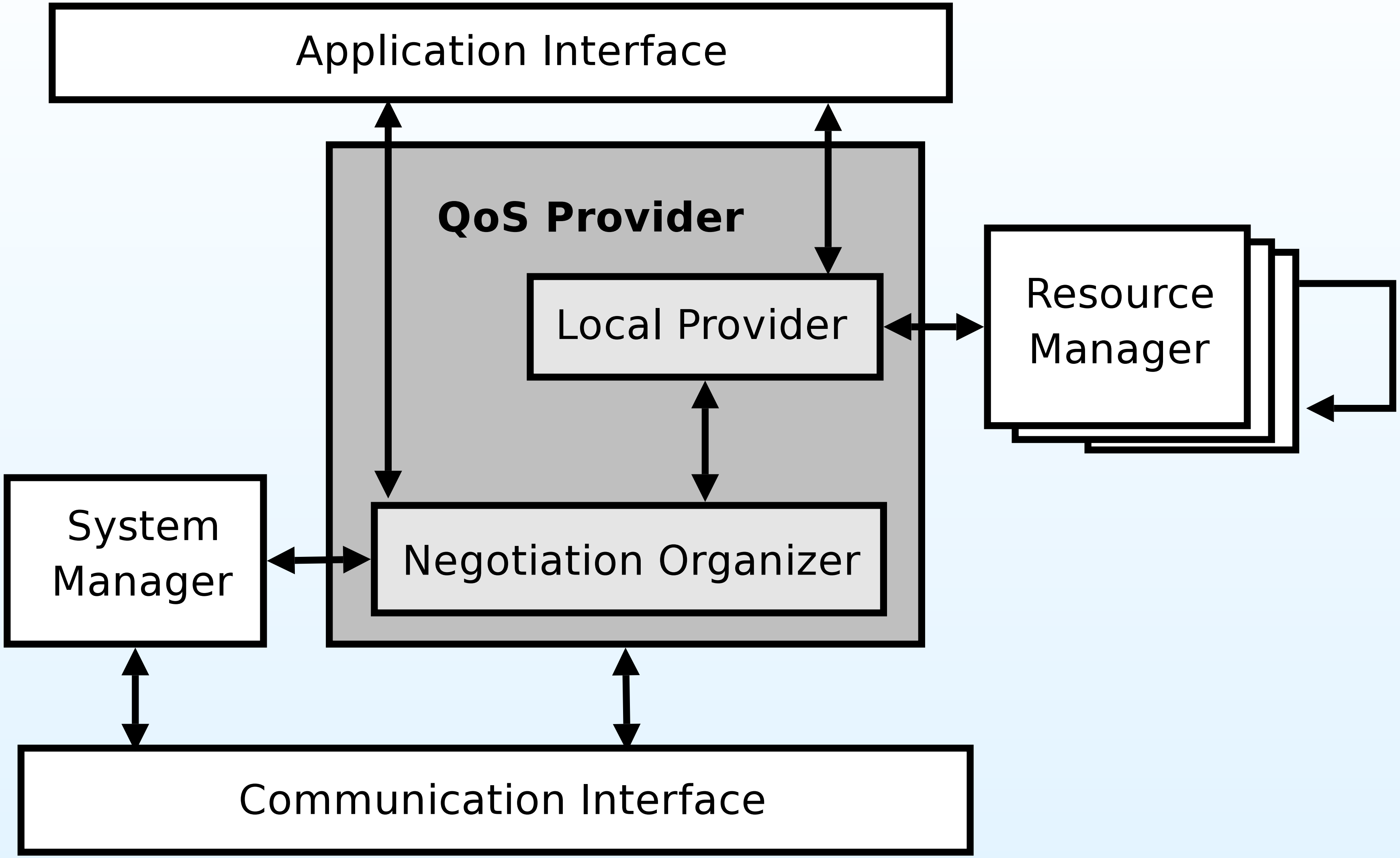
Problem statement

- Heterogeneous environment
  - Resource constrained wireless devices
  - High performance neighbors
- Several work in computation offloading
  - To face heavy resource requirements in clients
- However
  - End-users do not have **real influence** on obtained QoS
  - **Different users** tolerate **different QoS levels** or combination choices
  - How to **select the best neighbors** according to user's QoS requirements?

Proposed approach

- User specify **spectrum** of acceptable QoS levels in **request**
  - Through semantically rich QoS specification interface
  - Relative decreasing order expresses user’s preferences
- Underlying offloading mechanism splits application into taks
- Formation of temporary **coalition** for service execution
  - Neighbors **cooperate** with resource-constrained device
  - Taking advantage of **global** available resources
  - **Offloading** computation

Framework



Application Interface
QoS Provider
Local Provider
Resource
Manager
System
Manager
Negotiation Organizer
Communication Interface

QoS specifi cation interface

- QoS is often **multi-dimensional**
- Important to provide a **semantically rich** QoS specification
  - Users specify acceptable QoS levels
  - **Quality tradeoff** when resources are scarce

$$QoS = \{Dim, Atr, Val, DA_r, AV_r, Deps\}$$

- Proposed scheme
  - **Defines** dimensions, attributes and values of a domain
  - **Relations** that maps
    - dimensions $\longrightarrow$ attributes
    - attributes $\longrightarrow$ values
  - **Dependencies** between attributes' values

Service request

- Conflict
  - Rich user's request $\longrightarrow$ accurate proposals' evaluation
  - User can't specify utility of every quality choice
- Impose a **preference order** over dimensions, attributes and values
  - Relative decreasing order expresses user's preferences
- Example:

1. `Video Quality`
   (a) `frame rate: {[10→5],[4→1]}`
   (b) `color depth: {3,1}`
2. `Audio Quality`
   (a) `sampling rate: 8`
   (b) `sample bits: 8`

Coalition formation

- Objectives
  - Enable **cooperation** between neighbors
  - Address **increasing demands** on resources and performance
  - Maximize **user’s influence** on QoS provisioning
- Different groups of nodes $\longrightarrow$ different service execution performance
- **Distributed** QoS optimization algorithm
  - **Evaluation** of multi-dimensional proposals
  - **Selection** of nodes offering service **closer** to user’s QoS preferences

Coalition formation

1. On the source node N_i, *QoS Provider* broadcasts **description** of each task T_i as well as user's **QoS constraints** Q_i
2. Every N_j **formulates proposal** and replies to N_i with proposal P_j and its **local reward** W_j, resulting from its proposal acceptance
3. *QoS Provider* at N_i **evaluates** all received proposals for each T_i and **selects** the own that offers the values **closer to user's QoS constraints** Q_i
4. N_i offloads each T_i to winning node(s)

Proposal's formulation in neighbor node

- Proposal's formulation centered in two principles
  - **User's QoS constraints** expressed in request
  - **Local reward** of accepting new task
- Local QoS Provider
  - Recomputes QoS levels for new set of local tasks
    - Maximizing local reward
  - May involve degrading some tasks
- Guaranteeing user's request
  - Receive service at **one** of requested QoS levels

Proposal's formulation in neighbor node

- Each task T_i have an associated set of user's preferences
  - Presented in decreasing relative order
- Each k QoS dimensions have n possible attributes

1. Start by selecting the best QoS level in all k dimensions, $Q_{kj}[0]$, for the new arrived task T_a
2. While the new set of tasks is not schedulable
   - For each task T_i receiving service at $Q_{kj}[m] > Q_{kj}[n]$
     - Determine the **utility decrease** resulting from degrading attribute j to $m + 1$
     - **Find** task T_{min} whose decrease is **minimum and degrade it** to the $m + 1$'s level

Local reward

- Degree of **global** satisfaction

$$r = \begin{cases} n & \text{if task is being served at } Q_{kj}[0] \text{ for all dimensions} \\ n - \sum_{j=1}^{n} penalty_j & \text{if } Q_{kj}[m] > Q_{kj}[0] \end{cases}$$

- *penalty* is a parameter that decreases the reward value
  - Increases with distance to preferred values

Proposals' evaluation in source node

- Relative decreasing order in user's request
  - Imposes preferences
- Proposals evaluated **according to user's preferences**

$$distance = \sum_{k=1}^{n} w_k * dist(Q_k)$$

- For each dimension k evaluate
  - **Difference** between **proposed** and **requested** values

$$dist(Q_k) = \sum_{i=1}^{attr_k} w_i * dif(Prop_{ki}, Pref_{ki})$$

Proposals' evaluation in source node

- Degree of **acceptability** of proposed value
  - Compared to requested one

$$dif = \begin{cases} \dfrac{Prop_{ki} - Pref_{ki}}{max(Q_k) - min(Q_k)} & \text{if continuous } Q_{ki} \\[2ex] \dfrac{pos(Prop_{ki}) - pos(Pref_{ki})}{length(Q_k) - 1} & \text{if discrete } Q_{ki} \end{cases}$$

Conclusions

- Resource-constrained nodes may need to cooperate
  - To fulfill services at user's QoS preferred values
  - Coalition's performance is superior
- Users have different QoS requirements
  - Expressed through a semantically rich QoS specification
- Distributed service allocation
  - Multi-attribute proposals' evaluation
  - Selecting nodes offering service closer to user's preferences
- Proposals formulation for service execution
  - Local QoS optimization heuristic