

Efficient Network Selection in Heterogeneous Communication Scenarios using Arbitration

Martin Wegner

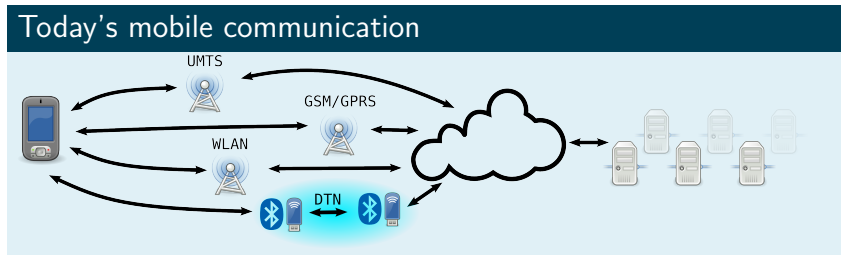
Institut für Betriebssysteme und Rechnerverbund
Technische Universität Braunschweig, Germany

June 17th, 2010

Content

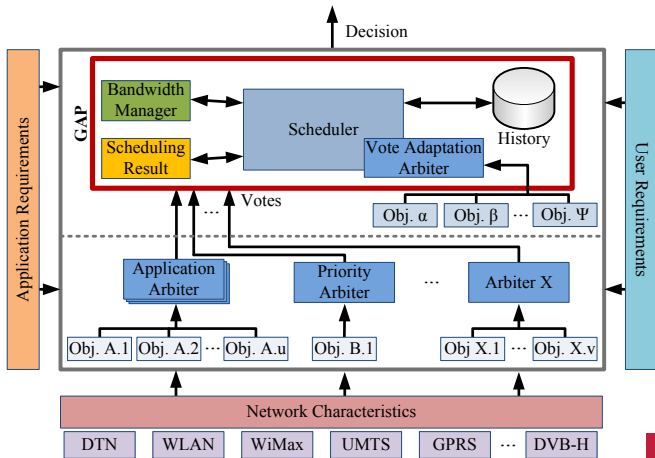
1. Introduction
2. Arbitration algorithm
3. Evaluation

Motivation



- ▶ Mobile devices can use a variety of different network connections.
- ▶ *But:* Usually only one network is used at a time.
- ▶ *Approaches mostly are:* always best connected, application-oriented, ...

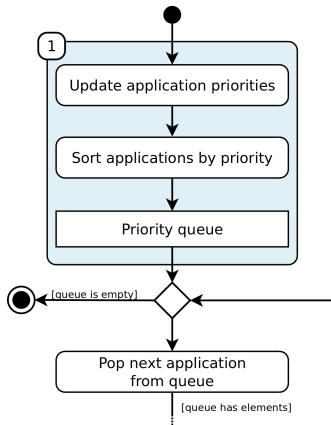
muXer architecture



Prerequisites

- ▶ **Votes** determine the eligibility of networks, they abstract from
 - ▶ network characteristics,
 - ▶ application requirements,
 - ▶ user requirements.
- ▶ **Decisions** are based on
 - ▶ votes,
 - ▶ application priority (e. g. OS priority),
 - ▶ bandwidth.
- ▶ **Possible decisions** are per application:
 - ▶ assign *single network* to application,
 - ▶ assign *bundle of networks* to application,
 - ▶ queue application for later consideration.

Decision process I: Prioritization

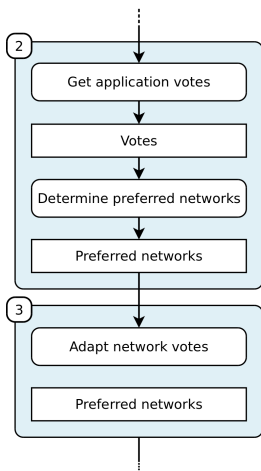


1. Prioritization:

- ▶ Priority is fetched and increased if application. . .
 - ▶ is interactive (fixed value),
 - ▶ is queued (and not currently active) (incrementally),
 - ▶ approaches previously set deadline (incrementally).

2. In order of this priority applications are queued and processed further on.

Decision process II/III: Pre-selection and adaptation



For each application:

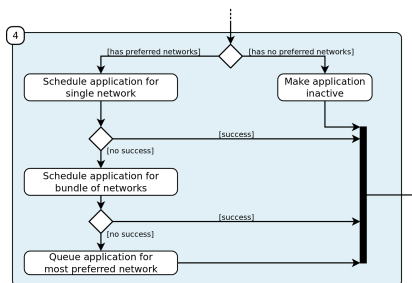
2. Selection of eligible networks:

- ▶ Fetch votes for all networks.
- ▶ Networks below certain threshold are not considered.
- ▶ Ordered by decreasing preference.

3. Votes (and network order) are adapted to...

- ▶ allow global optimization, e. g. to prefer networks already in use,
- ▶ implement network affinity (to avoid ping-pong effects).

Decision process IV: Network assignment



4. Assignments rely on successful bandwidth reservation in component *bandwidth manager*.

Subsequently try to assign

- ... to a single network;
- ... to a starting bundle of two most preferred networks, then incrementally adding one left network.

Otherwise, queue for most preferred network.

Decision process: Result and (re-)iteration

The **result** consisting of

- ▶ one *list of inactive applications*,
- ▶ per network or bundle:
 - ▶ list of *active* applications,
 - ▶ list of *queued* applications

is passed to the component *path management* for re-configuration.
Continuously repeated runs may be triggered. . .

- ▶ either *time-based*, e. g. a fixed interval
- ▶ or *event-based*, e. g. new applications, finished data transmissions, etc.

Evaluation: Setup

To assess the *overall quality* of the implemented scheduler, a *comparitive benchmark* was used.

- ▶ Various scenarios were randomly generated and simulated:
 - ▶ Variable number of networks and applications.
 - ▶ Optimum bandwidths between 53.6kbit/s (GPRS) and 54Mbit/s (802.11g).
 - ▶ Available bandwidth between 10kbit/s and correspondent optimum.
 - ▶ Varying application requests (bandwidth, interactivity, ...).

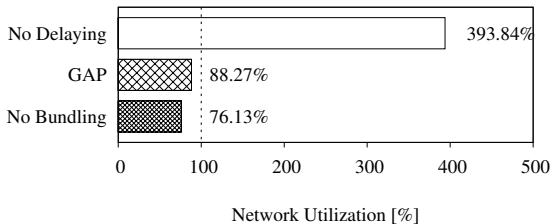
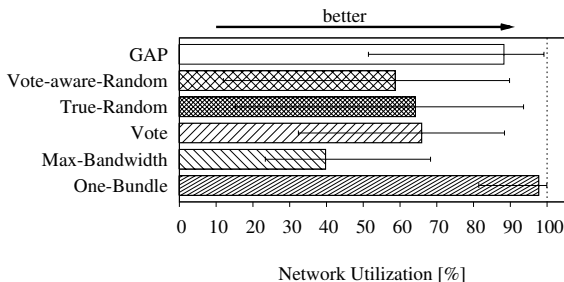
⇒ All varied for 100,000 runs.
- ▶ Strategies to compare to:
 - ▶ Random scheduler
 - ▶ Random scheduler considering only networks over threshold
 - ▶ Just assigning to network with highest available bandwidth
 - ▶ Always assigning most preferred network
 - ▶ Assigning to bundle of all networks

Metrics

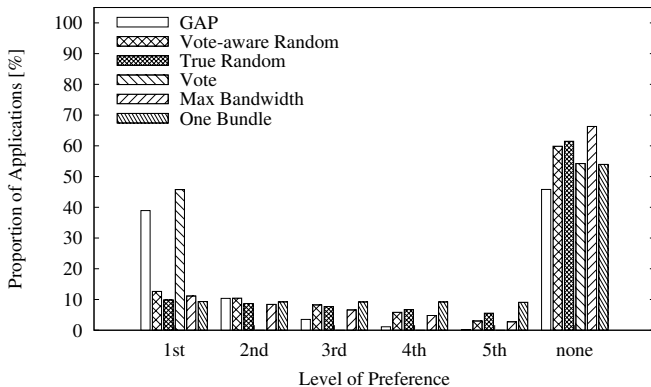
Analyzed metrics include:

- ▶ *Efficiency*
 - ▶ Overall network utilization
 - ▶ Utilization of active networks
 - ▶ Number of active networks
- ▶ *Quality*
 - ▶ Placement of network in application ratings
 - ▶ Number of active applications
- ▶ *Effect of bundling and delaying*
E. g. network utilization. . .
 - ▶ with bundles deactivated
 - ▶ with queueing deactivated
- ▶ *Scalability*

Exemplary results I



Exemplary results II



Thank you for your attention!
Are there any questions, please?