Re-scheduling with temporal and operational resources for the mobile execution of dynamic UMTS applications

Roman Englert

Rheinische Friedrich-Wilhelms-Universität Bonn

Institute of Computer Science

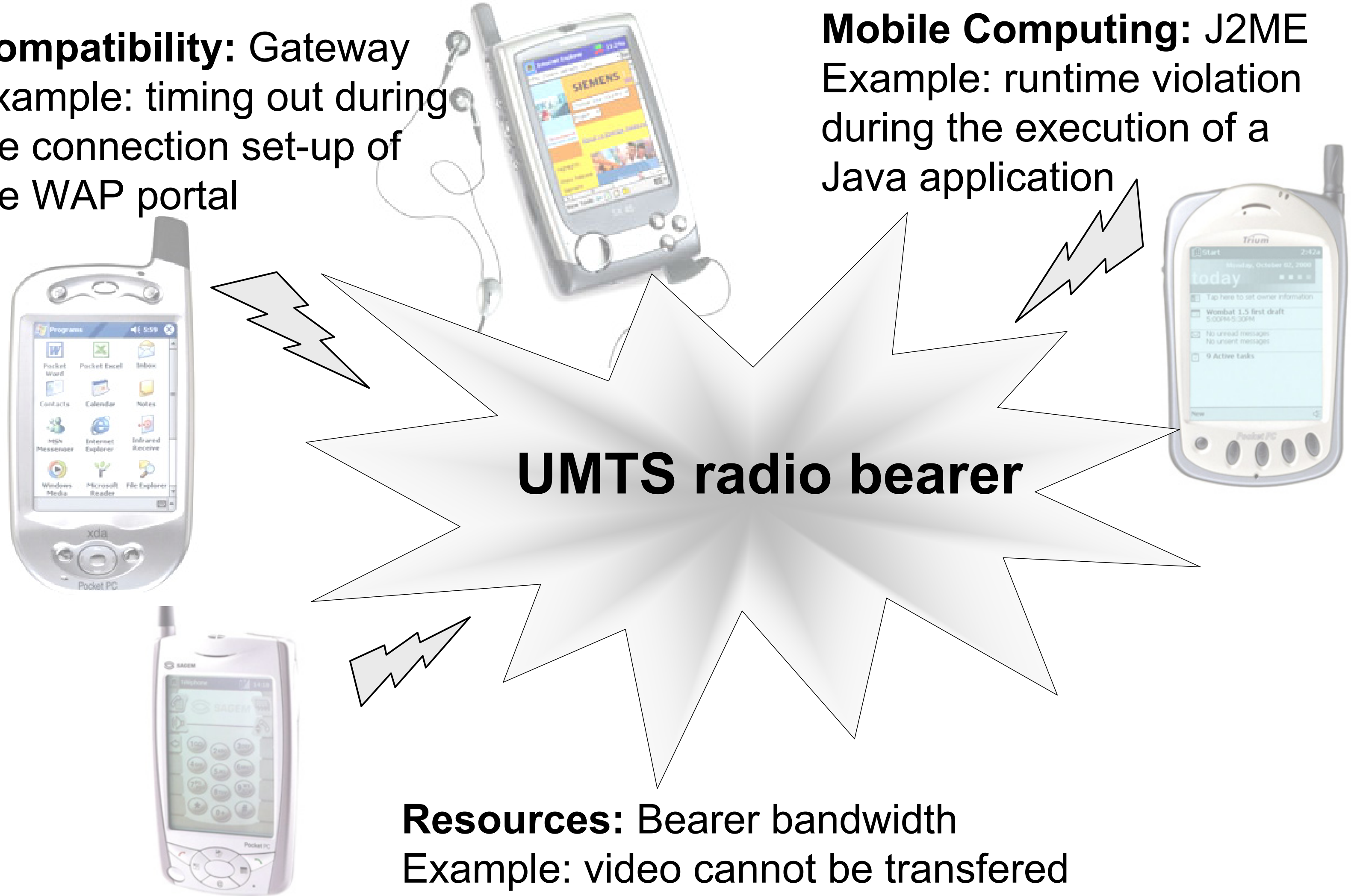
Germany

Hamburg, September 2003

Agenda

- Challenges and goal
- UMTS call set-up
- Quality of Service
- Planning and enhancements
- Experiments
- Implementation in the radio network

Challenge and goal



Compatibility: Gateway
Example: timing out during the connection set-up of the WAP portal

Mobile Computing: J2ME
Example: runtime violation during the execution of a Java application

Resources: Bearer bandwidth
Example: video cannot be transfered

Challenge and goal

- Limited resources in mobile terminals make a sequential execution of applications necessary.

→ As a consequence, a user has to wait a disagreeable waiting period until all desired applications are started.

- Core of the application start is the UMTS call set-up (based on circuit-switched technology; continuous signalling).

→ **Idea**: Modularize the call set-up and optimize the call set-up execution of several applications by scheduling.
As a result, mobile applications are executed immediately more or less in parallel after their initiation. See illustration on next page…

- **Real-life condition**: Users start applications one after the other and that leads to a dynamic world scenario, where applications do not have a coincident start point.
The dynamic world scenario requires re-scheduling.
- **Solution**: plan completion of incomplete plans.

Challenge and goal

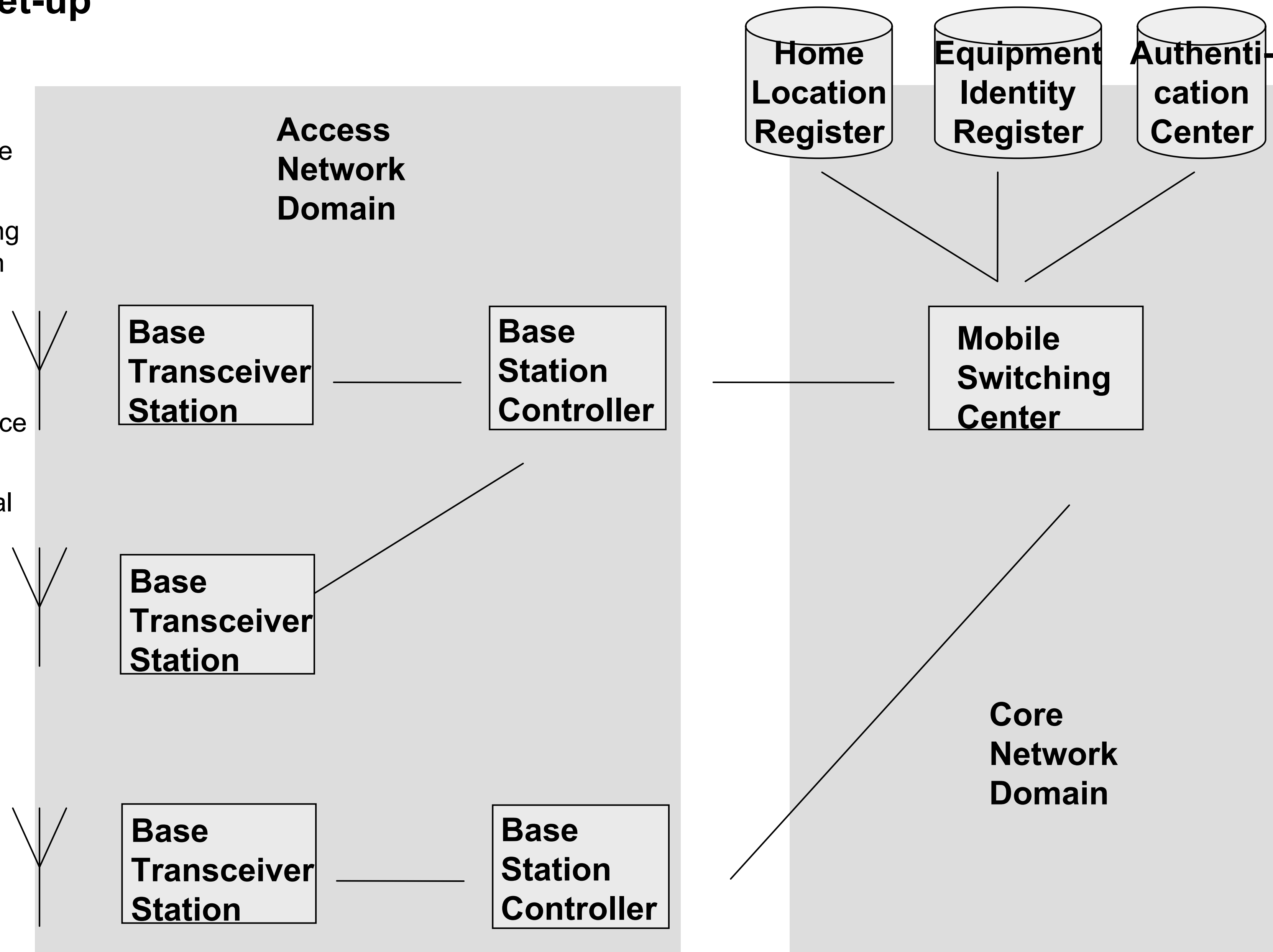
Market price ticker (Mt) Chess game (Cg)

Mt_{TRM} Mt_{CT} Cg_{TRM} Mt_{RRC} Cg_{CT} ...

UMTS call set-up

Steps:

- Terminal resource management
- Connection Timing & Authentification
- Agent-based logical resource booking
- Physically resource booking
- Mapping of logical and physical resources
- Bearer establish-ment
- Call set-up successfull



Discrete UMTS call set-up modeling

Discrete modeling of call set-up is possible via agents.
This leads to the modularization of the UMTS call set-up.

Agent Management
Requirements of mobile applications are transferred to bearer, e.g. QoS, required data volume, ...

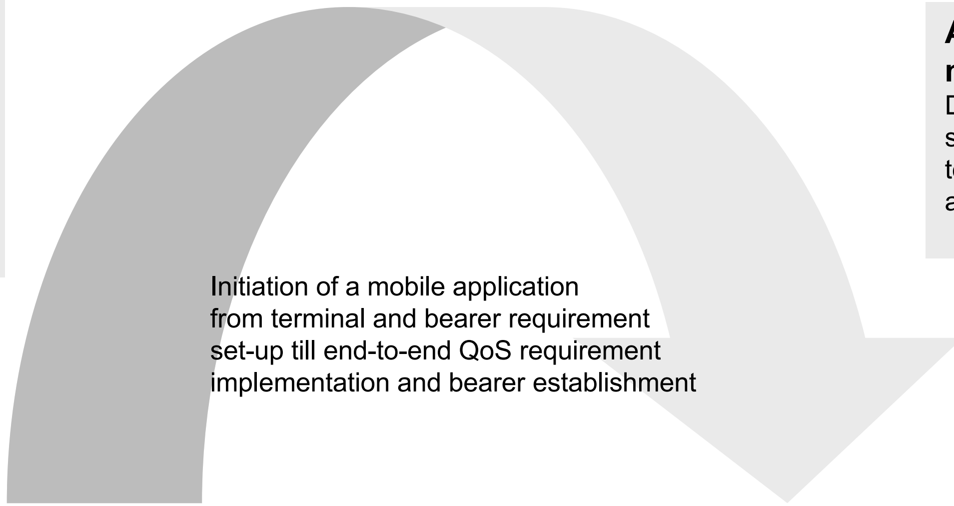
Agent Execution Environment Management
Information about mobile application are sent to AM, e.g. required servers and PDN, ...

Radio Resource Controler
Allocation of QoS by logical resources (mapping on MAC level in bearer)

Radio Allocation Bearer
Bearer allocation of QoS and in case of failure initiation of resource negotiation with mobile terminal

Connection Timing & Authentification
Connection set-up duration is monitored in the bearer and in case of failure feedback to the terminal is given (within a certain time, e.g. 1 sec.)

Agent Execution Environment Internet
Data transfer for application set-up from mobile terminal to core network and PDN, and vice versa



Terminal Ressource Management
An application start follows the resource availability check in the mobile terminal and the resource allocation

Bearer Service
Bearer establishment and feedback to mobile application, TRM and AEEI

Modules are executed in sequential order

Quality of service classes: taxonomy of applications

Traffic class	Conversation	Streaming	Interactive	Background
Fundamental Characteristic	Preserve time relation (variation) between information entities of the stream Conversational pattern (stringent and low delay)	Preserve time relation (variation) between information entities of the stream	Request pattern response Preserve data integrity	Destination is not expecting the data within a certain time Preserve data integrity
Example	voice, videotelephony, video games	Streaming multimedia	Web browsing, network games	Background download of emails

Holma/ Toskala (2000): page 12

Quality of service parameters

Maximum bitrate (kpbs): delivered between UMTS networks/ applications.
Application: Code reservation.

Guaranteed bitrate (kbps): Minimal required bitrate.
Application: Eases *admission control.*

Delivery order (y/n): delivery of Service Delivery Units (SDU) in sequentiell order of the input stream or not.
Application: Setting of the User Protocol. As an effect all applications can be executed with the same priority.

Maximum SDU size: maximal acceptable SDU size.
Application: Eases *admission control.*

SDU format information (bits): provides list of possible SDU applications (supports the spektral efficiency (bearer resource optimization).

SDU error ratio: determines ratio of erroneous SDUs.
Application: Configuration of protocols, algorithms, ... in the UTRAN

Transfer delay (ms): maximum delay for the 95-percentile of the SDUs.
Application: delay for the applications.

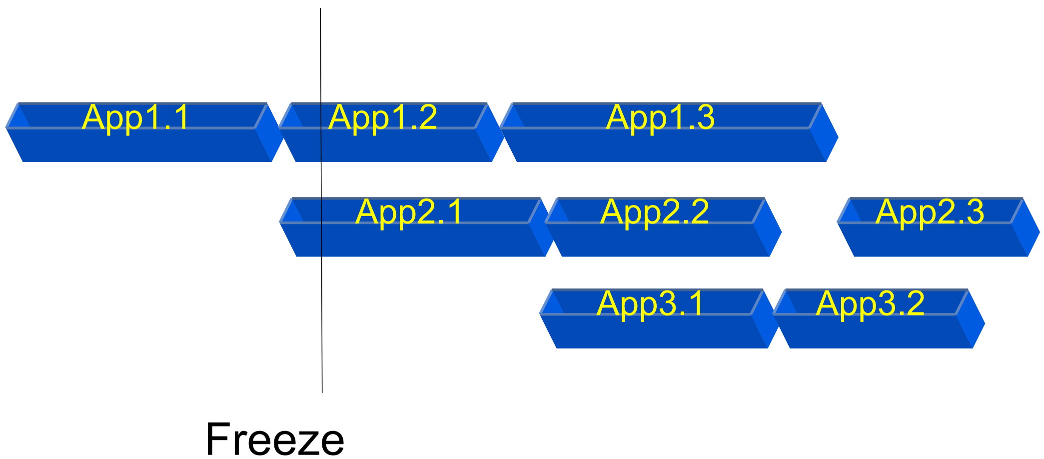
Traffic handling priority: relative importance of the SDUs.
Application: differentiation of bearers for classes, e.g. *interactive*

→ These parameters determine QoS. They denote resources within the scheduling process.

Planning and enhancements

Dynamic scenario:

- Assume application 1 and 2 are in the execution process and a new application 3 is initiated.
- Freeze execution state -> re-planning.
- Already executed modules are not considered for re-planning
  ➔ significant reduction of re-planning effort.



Note: Most plan repair methods are not applicable, since they are based on the removal of tasks (modules) and a subsequent plan repair.

➔ Plan completion...

Planning and enhancements (a more formal consideration)

A **complete plan** is a temporal ordering of actions, where each variable of the actions is bound to a constant.

Example: trm(app_1) ^ trm(app_2) ^ ct (app_1) ^ am(app_1) ^ ct(app_2) ^ am(app_2).

Temporal ordering can be partially specified via the BEFORE constraint that is defined by the binary predicate A BEFORE B, with actions A and B:

The **BEFORE constraint** specifies actions A and B such that in any plan action A must be executed and finalized before action B (there may be other actions after A and before B).

Example: trm(app_1) BEFORE ct(app_1) ^ ((trm(app_1) BEFORE trm(app_2)) v (trm(app_2) BEFORE trm(app_1)).
In the above example the modules trm of both applications cannot be executed in parallel.

Example: For the UMTS call set-up two kinds of constraints occur:

Intra-application orders the modules of one application, e.g. trm(app_1) BEFORE ct(app_1).

Inter-application orders modules with same names of different applications, e.g. trm(app_1) BEFORE trm(app_2).

Planning and enhancements (a more formal consideration)

A **incomplete plan** is a set of actions whose temporal order may be incompletely specified by ordering constraints.

Example: trm(app_1) ^ trm(app_new) ^ ct(app_new) ^ am(app_new) ^ trm(app_2) ^ ct (app_1) ^ am(app_1) ^ ct(app_2) ^ am(app_2).

In this example app_new is started after module trm of application 1 is executed. As result several BEFORE constraints are violated, e.g. ct(app_1) BEFORE ct(app_new).

Plan repair is done by the completion of incomplete plans:
Assume the incomplete plan P is a finite set of complete plans *T*. Then each complete plan t in *T* is a **completion** of P, if *T* imposes the ordering constraints of P.

Example: Let P be {trm(app1) ^ trm(app2) ^ ct (app1) ^ am(app1) ^ ct(app2) ^ am(app2)} U *T'* with *T'* = {trm(app_new) ^ ct(app_new) ^ am(app_new)}, where the union of plans means their concatenation with temporal order preservation.

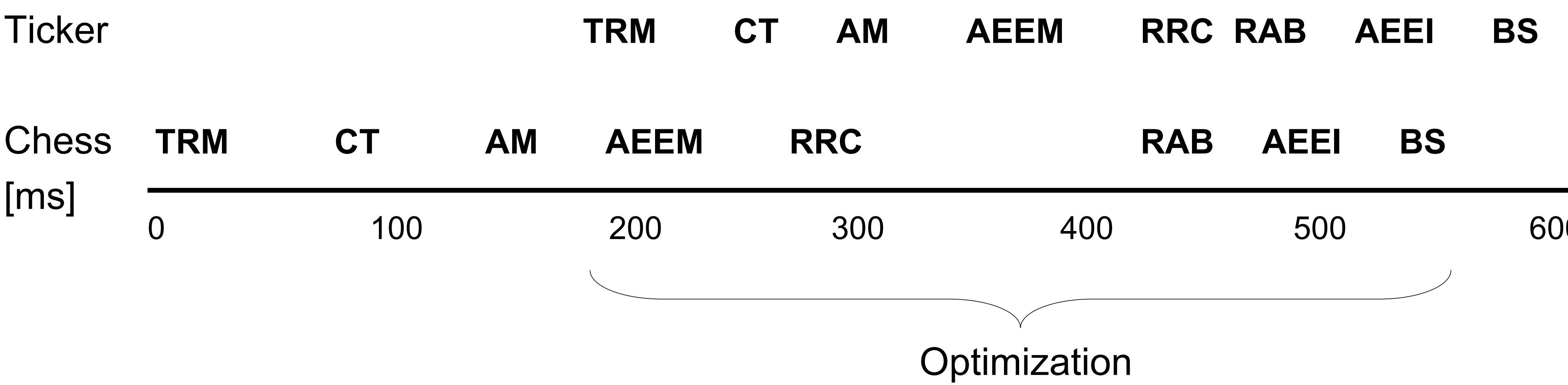
Then *T'* is a completion of P, since the BEFORE constraints can be fulfilled.

Experiments

Scheduling domain consists of

- Modules and execution duration times for each application
- Resources (QoS parameters)
- BEFORE, Intra- and inter-application constraints

Two applications on one mobile applied to the temporal- and resource-based planner TP4:



Experiments

What are the questions to be answered in the dynamic case?

- **Real-time:** Can plans for the execution of mobile applications be completed in an appropriate time? Plan completion has to be done with a maximum duration that does not exceed the original plan execution time.
- **Completeness:** Is it possible to complete the incomplete plan, i.e. does plan completion result in an (optimal) plan for the required applications that minimizes the waiting period until all applications are started?

N-1 applications and 1 new app.	2	4	6	8	10
time [ms]	0.01	0.03	0.04	0.06	0.10

- Evaluation:
  - The execution of a call set-up takes approximately some seconds for an interactive game and up to 30 seconds for a WAP connection.
  - The re-planning effort is much shorter: it takes only 0.01 up to 0.1 seconds for 10 applications, assuming that the instances for actions have been pre-computed as TP4 does.
  - If the execution of a call set-up module results in a failure then the application has to be restarted and the waiting period for the user until the other applications are started will decrease due to the re-planning.

Implementation in the radio network

Why is the approach feasible in the sense that it can be implemented in a radio network?

Terminal Equipment Domain (TED)

Mobile + Fixed

Uu: GC, Nt, DC

Access Network Domain (AND)

Examples:
- DECT
- LAN
- ISDN
- B-ISDN
- GSM
- UMTS

Iu: GC, Nt, DC

Core Network Domain (CND)

Examples:
- GSM + IN
- ISDN + IN
- B-ISDN + IN
- B-ISDN + TINA
- TCP/ IP

Implementation in the radio network