Simonstrator-Platform

Simulation and Prototyping Platform for Distributed Mobile Applications

Björn Richerzhagen, M.Sc.
Bjoern.Richerzhagen@kom.tu-darmstadt.de
Internet-based services delivered to mobile devices
  - Live / on-demand video
  - Location-based services
  - Collaborative games

Often in challenging communication environments
  - Rapidly changing signal quality
  - Interrupted connectivity
  - Context-dependent utilization
    - Location, time, intention, …

→ Need for adaptive systems
Example: Adaptive Video Streaming System

Bandwidth changes
- Multiple video representations (quality) and an adaptive player

Connectivity loss
- Pre-fetching content
- Utilizing local delivery (ad hoc)

Load patterns
- Pay a CDN or aid with P2P-based delivery of streams
- Utilize SDNs for multicasting to larger groups within an ISP
Distributed application protocol

- Peer-assisted or P2P-based
- 1k-10k devices, focus on coordination among devices

Network-centric effects (L1-L4)

- Local dissemination (BT/Wi-Fi)
- Utilization of SDN technologies

User-centric effects

- Perceived video quality
- Application usage patterns

→ Evaluation required on all scopes; scopes affect each other
Our Approach

The Simonstrator-Platform in a Nutshell

Wish: use a **single** implementation of the system for **all evaluation scopes**

Rely on existing simulators and their (verified) models

- User movement, network protocols, wireless transmission

Enable easy deployment on PCs and Android mobile devices

- Access to device-specific APIs
- Unified instrumentation concepts
- Specifically targeted towards mobile distributed applications
Outline

Motivation
- Distributed mobile applications
- Different evaluation scopes

The Simonstrator-Platform
- Overview and concept
- Core framework
- Instrumentation
- Runtime environments

Current use cases

Conclusions & Outlook
Core framework
- Component interfaces
- Scheduling and instrumentation

Runtime environments
- Network simulators
- Standalone PCs and Testbeds
- Android Smartphones

(Your) applications and services
- Composed out of multiple framework components
Component-based Composition

All components are accessible via the host container

- Using their (super-) interface

Components are provided by the runtime environment or as exchangeable libraries

- Transparent for the application
- Enables experiments with different grades of abstraction
- Allows rapid, “stub-based” development in larger projects

```java
host.getComponent(LocationSensor.class);
```
The Simonstrator-Platform

Instrumentation

**Metrics (pull)**
- Apps annotate local state (e.g., a buffer state or # neighbors)
- State is accessible via a unified interface prov. by the framework
- Integrated processing toolchain (plotting, persistence) and bridge to common network simulators

**Analyzers (push)**
- Application-specific analyzers (e.g., onActionX)
- Implementation can optionally be provided by the runtime
Implementations of (a subset of) component interfaces

- Minimal set is Scheduling, Times, Randoms, Net I/O
- Event-based programming model known from network simulators
- Net I/O uses a shared library for interoperability between runtime environments

Current runtime environments

- Overlay simulator PeerfactSim.KOM (done)
- Android app and service (done)
- Java standalone (done)
  - Network simulator OMNeT (work in progress)
  - MaxiNet network emulator (work in progress)
  - WARP SDR (work in progress)
Outline

Motivation
- Distributed mobile applications
- Different evaluation scopes

The Simonstrator-Platform

Current use cases
- CRC MAKI
- Live video streaming
- Augmented reality gaming

Conclusions & Outlook
18 professors, ~30 researchers, organized in 10 sub-projects

Fostering collaboration by providing a common platform

- Integrate individual research prototypes into overall toolchain
- Build a common demonstrator to showcase research results
- Enable user studies and evaluations on real devices

Use cases: Video Streaming and Event Dissemination
Use Cases – Live Video Streaming

Simulation Study of a P2P Streaming Overlay

P2P Streaming Overlay “Transit”

- Adapts to current load by executing mechanism transitions
  - Delivery method and topology is switched during runtime
- Supports content adaptation with Scalable Video Codec (SVC)

Simulation study [1, 2]

- Up to 5k nodes in total
- During flash crowd: arrival rate increases to 800 nodes / minute
- 3 video quality representations

Use Cases – Live Video Streaming

Realization using the Simonstrator-Platform

Streaming component

- Offers chunk-based interface (Video Block)

Video model / video block

- Block size (bytes) is annotated
- Payload is included depending on the runtime environment

Transport of video blocks

- Size-based in simulations (no serialization, lower overhead)
- Real video data on prototypes
Use Cases – Live Video Streaming

Resulting Prototype on Android

Java-based streaming server
- Simulation model of “Transit”: source and tracker component
- Running within the Standalone Runtime Environment
- Interfacing VLC for playback and control of the video stream

Android-based client application
- Integrating the “Transit” overlay without code modifications
- Real video data is streamed and shown on the handheld devices

Use Cases – Live Video Streaming

**Ongoing: Software-defined Multicast (SDM)**

Network-centric evaluation scope
- ISPs offer the streaming system a rentable multicast-service
- Focus on network-centric metrics (e.g., flow rules, throughput, ...)

SDM in the Simonstrator
- Provide a SDM-component that can be utilized by the overlay
- Realization of SDM within the MaxiNet runtime environment
- Simple simulation model for the PeerfactSim.KOM environment

Players’ actions in the real world are relevant to nearby users

- In-game interaction with physical objects (Points of Interest)
- Players collaborate in groups

Adaptive event dissemination with context-based pub/sub

- Can utilize ad hoc connectivity between nearby devices
- Supports different local protocols
- Adaptations controlled by central game service (cloud-based)

Use Cases – Augmented Reality Gaming

Evaluation Scopes of “Bypass”

Simulation Study
- Scalability in the central control alg.
- Synthetic movement and load models
- Focus on the event dissemination

User Study
- Measuring the behavior of players
- Testing feasibility of local dissemination
- Real ( playable) game prototype

Demonstration
- Showcasing local dissemination
- Integrating Bluetooth and NFC
- Scaled-down version of prototype

Outline

Motivation
- Distributed mobile applications
- Different evaluation scopes

The Simonstrator-Platform

Current use cases
- CRC MAKI
- Live video streaming
- Augmented reality gaming

Conclusions & Outlook
Enable “multi-scope” evaluations of mobile distributed systems

- Relying on well-established simulation models and platforms
- Providing component-based access to platform APIs
- Android (Java) as target platform

Rapid integration of research prototypes in large projects

Platform (beta) via git available online: https://goo.gl/p3q9VP
- Contact us for full access!
Thank You!

Git-Repository: goo.gl/p3q9VP

www.kom.tu-darmstadt.de
www.maki.tu-darmstadt.de

Björn Richerzhagen, M.Sc.
The Simonstrator-Platform

Scheduling and Network I/O

Event-based programming model
- Used by most simulators
- Scheduling

Network I/O can be extended
- SDR: allow control of MACPHY properties within an application
- Mixed setups due to compatible binary data format of messages