Federating OMNeT++ Simulations with Testbed Environments

Asanga Udugama\textsuperscript{1}, Koojana Kuladinithi\textsuperscript{2}, Anna Förster\textsuperscript{1} and Carmelita Görg\textsuperscript{1}

\textsuperscript{1} - Sustainable Communication Networks Group, University of Bremen, Germany
\textsuperscript{2} - Institute of Communication Networks, Hamburg University of Technology, Germany

2\textsuperscript{nd} OMNeT++ Community Summit 2015,
Zurich, Switzerland – September 3-4, 2015
Contents

• Keetchi Architecture
• Codebase Architecture
• OMNeT++ Model
• Simulated Scenarios and Evaluations
• Summary and Future Work
Keetchi Architecture
Features of Keetchi

• Information centric communications
• Distributed control
• Distributed caching
• Reinforcement learning based data handling
• Opportunistic data propagation
Operation

- Messages: Data and Feedback

![Diagram showing data and feedback flow between devices labeled Alice, Bob, and Charlie. Data moves from left to right with timestamps and feedback indicated by arrows.]

- An Keetchi deployed device
- Cached data item
- Data as blue arrows
- Feedback for data (i.e. red sofa) as orange arrows

---

Universität Bremen
Codebase Architecture
Requirements and Architecture

- **Requirements**
  - Test-bed devices
  - OMNeT++ simulations

- **Architecture**
  - Platform dependent API
  - Keetchi functionality API
Exposed Functionality

- Incoming message processing
- Opportunistic message generations
- Status information servicing
- Wiring for statistics

```c++
/**
 * Process an incoming Feedback message from either the application
 * layer or the link layer.
 * 
 * @param fromWhere   The message source, application layer or link
 *                    layer.
 * @param feedbackMsg The Feedback message with its contents as a
 *                    KLFeedbackMsg instance.
 * @param currentTime The current clock time.
 * @return            A KLAAction instance with all actions required
 *                    to be taken by the Keetchi layer.
 */
KLAAction* processFeedbackMsg(int fromWhere, KLAFeedbackMsg *feedbackMsg,
                             double currentTime);
```
OMNeT++ Model
Model Considerations

• Keetchi Architecture
  • 3 Layer Protocol Stack, Keetchi functionality API

• Leverage of existing OMNeT++ Models
  • Mobility models, Traffic models, Wireless propagation models

• New Features
  • Energy and expenditure model
  • WiFi Direct and Bluetooth Low Energy
Simulated Scenarios and Performance Evaluations
Simulated Scenarios (Applications)

- Many scenario areas
  - e.g., emergency services, social networking, etc.

- *UniRecycler* application (“grapevine”)
  - peer-to-peer, decentralised and distributed communications
  - caching, opportunistic communications (store-and-forward)
Performance Evaluations

- Feedback system
- Opportunistic data propagation strategies
- Caching
- Mobility patterns
Summary and Future Work
Summary and Future Work

• Summary
  • Design and development of a communications architecture (Keetchi)
  • Development of a common code base for test-bed devices and OMNeT++ simulations

• Future Work
  • Inter-working with test-beds and simulation
Thank You.

Questions.