UMIC: Wie Exzellenzinitiativen mobile Netze umgestalten (wollen)

Otto Spaniol, Martin Krebs, Alexander Zimmermann

Lehrstuhl für Informatik 4
RWTH Aachen
Ahornstraße 55
D-52074 Aachen

spaniol@cs.rwth-aachen.de
www.nets.rwth-aachen.de
Outline

1. About UMIC
2. Wireless Mesh Networks: A new network technology
3. The UMIC-MESH.net wireless mesh testbed
4. Service discovery as an application
5. The “happy end”

More information: See  www.umic.rwth-aachen.de
About the Research Cluster UMIC

- **UMIC:**
  - Ultra High-Speed Mobile Information and Communication
- One of the 37 excellence clusters in Germany
- Funding period 10/2006-10/2011
- 6,5 Mio € per year plus overhead
- The cluster is organised in four research areas
Cluster Structure

- Area A: Wireless Transport Platform
- Area B: Mobile Applications & Services
- Area C: Radio Frequency Subsystems & System-on-Chip (SoC) Design
- Area D: Cross Disciplinary Methods & Tools
The Research Areas

A: Wireless Transport Platform
- New solutions for infrastructure, networks, radio links and terminals to get information and communication from/to mobile users

B: Mobile Applications & Services
- Representative classes of applications and services
  - Common requirements of each class
  - Potential trade-offs to match requirements of mobile applications and services with wireless transport platform

C: RF Subsystem and SoC Design
- Innovative hardware architectures leveraging opportunities and addressing challenges of future silicon technologies
- Design methods, tool concepts and programming for heterogeneous many-core systems-on-chip
The Research Areas

D: Cross Disciplinary Methods and Tools

- Mobile Applications and Services
- Wireless Transport Platform
- RF Subsystems and SOC Design
2. Wireless Mesh Networks
What are wireless mesh networks?

- **Backbone mesh gateways**
  - Wired internet backbone

- **Backbone mesh routers**

- **Routing mesh clients**
  - Wired connection

- **Non-routing mesh clients**
  - Wireless connection
Similarities with Mobile Ad-hoc Networks (MANET)
- Wireless, i.e. limited transmission range, high loss rates
- Multi-hop
- Redundancy
- Mobility
- Dynamics; self-organising, self-healing, self-configuring

Differences between WMNs and MANETs
- Pre-existing infrastructure
- Integration of heterogeneous networks
How to deal with Wireless Mesh Networks?

- Possible approaches
  - Mathematical analysis
  - Simulation
  - Emulation
  - “Virtualisation” (see next slide)
  - Real Testbeds

- Problems
  - Interpretation of results with regard to
    - applicability
    - repeatability
    - maintainability
    - scalability
    - scenario creation
Excursus: Virtualisation

- **System virtualisation**
  - the virtual machine simulates the complete hardware, allowing the execution of an unmodified “guest” OS for any CPU type.

- **“Paravirtualisation”**
  - the virtual machine does not need to simulate the hardware but offers a special API that can only be used by modifying the “guest” OS.

- **Virtualisation on “operating system level”**
  - this concept enables multiple isolated and secure virtualised servers to run on a single physical server. The “guest” OS environments share the same OS as the host system.
3. UMIC-Mesh.net wireless mesh testbed
Network Protocol Development

- Realisation of networking protocols and tools
  - Iterative process
    - Developing: Implementation and debugging
    - Distributing: Installation and validation
    - Testing: Functionality and performance
  - In case of failures: debugging information has to be collected and analysed

- Problem
  - Iterative software development process is time consuming in particular when new software versions have to be distributed
A hybrid testbed

Goals

- Benefit from the advantages of a real testbed
- But avoid its disadvantages (unrepeatability, uncontrollability)

How can we reach these goals?

- **Hybrid testbed**: real testbed plus virtualised environment
- Distribute the tasks of network protocol development
  - Virtualised environment for developing, validating and testing of the functionalities
  - Real testbed for performance evaluation
- No need for an accurate emulation of a wireless medium

3. UMIC-Mesh.net Testbed
Maintenance process of a testbed

- **Problems**
  - High maintenance effort in particular for the real testbed (to a lesser extent for the virtualised environment)
  - A failure during the performance evaluation results in a labour-intensive distribution of a new protocol software version

- **Solutions**
  - **Central configuration:**
    - **Source functionality:**
      Provision of a single Operating System, Internet access
    - **Sink functionality:**
      Logging and measurements results
  - **Virtual backbone network**
3. UMIC-Mesh.net Testbed

- 51 mesh routers
- 6 of them for testing purposes
UMIC-Mesh.net routers
4. Service discovery as an application
What is service discovery?

Service request: “Is there an Irish pub in the vicinity?”

Service discovery describes the process of offering and finding services in a network without a-priori knowledge about the services.
Current proposals for service discovery do not scale in Wireless Mesh Networks:
- Simple flooding produces large overhead
- May need additional multicast routing support
- Central server solutions are unsuitable (bottlenecks, single point of failure, …)

Approaches for MANETs don’t consider mesh properties, i.e. no consideration of:
- Mix of routing and non-routing clients
- Combination of static backbone and MANET part
- Hierarchical structure
- Different mobility patterns

A unifying framework for different scenarios is necessary!
Our approach for the mesh backbone:
- Cross-layer approach for mesh routers
- **Proactive** Optimised Link State Routing (OLSR) with encapsulated multicast DNS based Service Discovery (mDNS-SD) – OLSR is used for example in Berliner Freifunk Netzwerk, MIT Roofnet,…

Our solution for the MANET part:
- Cross-layer approach for routing clients
- **Reactive** Dynamic MANET On-demand Routing (DYMO) with encapsulated multicast DNS based Service Discovery

Setup for non-routing clients:
- Unmodified multicast DNS based Service Discovery
Integration of OLSR and mDNS

- Encapsulate service discovery messages into OLSR packets
- Service discovery application sends messages to its local routing daemon and vice versa
- Advantage of packet encapsulation (a packet can contain several service discovery messages):
  - OLSR daemon controls packet handling
  - Use OLSR’s “advanced flooding techniques” (Multi Point Relaying)
  - Hop control for service requests and advertisements
- This protocol has been implemented as a prototype within the UMIC-mesh testbed
Scenario A: service discovery for non-routing clients

- A Service Proxy encapsulates service discovery messages into OLSR packets and caches “service history”

mDNS = multicast Domain Name System
Scenario B: service discovery for routing clients

- Routing clients execute the service discovery application and an OLSR Daemon
- Reactive routing protocols such as DYMO can also be used
- Given: A Wireless Mesh Network with a number of backbone service caches
- Problem: After one query for a service class most caches contain redundant services due to long-term usage and limited mobility
- Synchronisation is necessary
- Solution: pseudo synchronisation
  - Suppress unnecessary replies and send only “delta set” (set of unseen information)
  - Proposed coordination algorithm: *Farthest Node Sends First (FNSF)*
Example: cache reply scenario

Many caches respond which leads to redundant information.

With FNSF: only one cache responds.

A client queries for a service.
Farthest Node Sends First (FNSF)

Details of FNSF:

- Nodes are **synchronised by their hop distance** from the querying client.
- The distance is obtained from query message by the hop count field.
- Each node draws a backoff time $f(h)$ ($h=$ hop distance) and delays answering correspondingly for $f(h)$ time units.
- The "farthest" node replies first (exactly speaking the node whose backoff time ends first).
- All other nodes suppress already seen service replies and send only the "delta set" (additional messages).
- Already seen or forwarded services are stored.
Possible variants for the backoff time $f(h)$:

$f(h)$ should decrease with $h$ (apart from the randomisation value $x_{\text{rand}}$)

- $f(h) = h_{\text{max}} - h + x_{\text{rand}}$
- $f(h) = \alpha \times \frac{1}{h} + x_{\text{rand}}$

$h$: current hop count of the received query
$h_{\text{max}}$: maximum TTL (Time-To-Live) in hops
$x_{\text{rand}}$: random value from an interval $[0...x_{\text{max}}]$
$\alpha$: scaling parameter
Performance results

- 48 mesh routers, 20 service caches, 17 services, single querying client

Trade-off between replies and average delay:
Instead of a conclusion: the happy end

Instead of the usually rather boring conclusions or “Thank you very much” flowery transparencies:

Here is the HAPPY END of my presentation!