TAMPEREEN TEKNILLINEN YLIOPISTO
TAMPERE UNIVERSITY OF TECHNOLOGY

ROMAN FLOREA

NETWORKING SOLUTIONS FOR INTEGRATED HETEROGENEOUS WIRELESS ECOSYSTEM

Master of Science thesis

Examiners: Prof. Yevgeni Koucheryavy
and
Dr. Sergey Andreev
Examiners and topic approved by the
Faculty Council of the Faculty of
Electronics and Communications Engineering
on 4th May 2016

ABSTRACT

ROMAN FLOREA: Networking Solutions for Integrated Heterogeneous Wireless Ecosystem
Tampere University of Technology
Master of Science thesis, 57 pages, 5 Appendix pages
June 2016
Master's Degree Programme in Information Technology
Major: Communication Systems and Networks
Examiners: Prof. Yevgeni Koucheryavy and Dr. Sergey Andreev
Keywords: traffic offloading, heterogeneous network, SDN, LTE, WiFi, 5G

This work targets at applying computer networking techniques to address challenges in modern wireless networks and in various environments built around these networks. The main focus of the work is on designing and implementing prototypes and demonstrators to support research in domains of heterogeneous networks (HetNets). These research domains include centralized radio resource management in emerging cellular network architectures, network assistance role in device-to-device (D2D) communications, and studying prospective services in these networks. Within the research group the author was tasked with designing network architectures and demonstrating certain connectivity and functionality interesting for the research. The author was responsible for modifying commercial off-the-shelf equipment to become suitable for target research scenarios, selecting network technologies to achieve connectivity requirements, deploying network architecture entities within the research group's cloud platform. For HetNet track, the primary goal was to design a platform that would mimic a device connected through a heterogeneous network, allowing researchers to experiment with traffic flow optimization in an environment close to the envisioned next-generation network architecture. Prototype solution and testbed were designed building on software defined network principles of automation, abstraction and software based flow switching, and were implemented using overlay networks and virtual network functions. Within D2D communications research, the task was to design architecture demonstrating feasibility of traffic offloading from infrastructure network to direct links. Prototype was implemented with automated routing control in overlay network. To demonstrate novel services enabled by advanced security frameworks, D2D platform was augmented and a new network application has been implemented, also suitable for wearable electronics.

PREFACE

This work concludes 3 years of the author's activity within the Department of Electronics and Communications Engineering, Tampere University of Technology, Finland. The results presented in this thesis developed from multiple collaborations with other researchers and constitute a part of larger scale scientific effort within the research group.

I would like to thank Dr. Sergey Andreev and Prof. Yevgeni Koucheryavy for initiating me to the academic world and acknowledge their guidance throughout my work at the department. Also this work wouldn't be possible without all the brilliant people whom I was fortunate to meet at the research group. Furthermore, my deepest appreciations go to my comrades in arms Adam Surák and Aleksandr Ometov for their friendship and collaboration in both scientific and normal lives.

Finally, this achievement would be meaningless without my beloved Anna, my parents and other family members.

This work was supported in part by Intel Corporation, the Academy of Finland (project "Empowering Secure, Private, and Trusted Network-Assisted Device-to-Device Communication"), and the Internet of Things Program of DIGILE, funded by Tekes.

Tampere, 23.5.2016

Roman Florea

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS AND SYMBOLS

3GPP	Third Generation Partnership Project
4G	Fourth Generation
5G	Fifth Generation
ANDSF	Access Network Discovery and Selection Function
AP	Access Point
BBU	Base Band Unit
BGP	Border Gateway Protocol
BS	Base Station
CAPEX	Capital Expenditure
COTS	Commercial Off-The-Shelf
CRRM	Cooperative Radio Resource Management
D2D	Device-to-device
GRE	Generic Routing Encapsulation
HetNet	Heterogeneous Network
IoT	Internet of Things
LPN	Low Power Node
LXC	Linux Container
MIMO	Multiple-Input and Multiple-Output
NAT	Network Address Translation
NFV	Network Functions Virtualization
OPEX	Operational Expenditure
OVS	Open vSwitch
PBR	Policy Based Routing
POP	Point Of Presence
PoC	Proof of Concept
QoS	Quality of Service
RAT	Radio Access Technology
RPM	RPM Package Manager
RRH	Remote Radio Head
RSA	Rivest, Shamir, Adleman
RSSI	Received Signal Strength Indicator
SDK	Software Development Kit
SDN	Software Defined Networking

SLA	Service Level Agreement
TUT	Tampere University of Technology
UE	User Equipment
URL	Uniform Resource Locator
VETH	Virtual Ethernet
VM	Virtual Machine
WFD	WiFi Direct
eSMLC	Evolved Serving Mobile Location Centre
ePC	Evolved Packet Core
eNodeB	Evolved Node B
mmWave	Millimeter Wave

1. INTRODUCTION

1.1 Motivation

The evolution of wireless communications over the last decades has brought us to a point where users demand "anytime, anywhere" high speed mobile Internet access, and have high expectations on service quality (in terms of bandwith, latency, service continuity, privacy, security, etc.). Completion of the fourth generation (4G) of broadband communication standards in 2011 has introduced drastic improvements to the wireless systems deployments in terms of capacity, energy efficiency, and quality of service. Year 2015 was the first time when 4G traffic exceeded third-generation (3G) traffic [1]. Recently, the share of 4G mobile connections was at 14% and 4G mobile data traffic reached 47% [1], while current research efforts are shifting focus towards elaborating solutions that would comprise what may be referred to as fifth generation (5G) wireless networks. First deployments of 5G systems are expected around year 2020, which would conclude a historical 10-year cycle for a generation of communications standards.

Even though it is not yet exactly clear what technologies would comprise a 5G network, the trends in communications engineering and user demands already shape our vision on network requirements. As global mobile traffic increased with 1.6 exabytes of data per month in year 2015 (which makes it a 74 percent growth [1]), service providers seek ways to fulfill the ever increasing user expectations on being connected to the Internet at any location, with the ability to enjoy all the multimedia services offered through the network at any time. These demands for uniform connectivity regardless of users' location, who their connection peers are, and what their preferred services are, pose significant challenges to the development of 5G technology to provide matching data rates.

However, modern wireless networks currently lack what it takes to provide experience of ubiquitous connectivity. Currently, the networks are unable to deliver

uniform data rates and are subject to excessive time delays, or even service disruptions due to coverage issues and influence of interference conditions. Current technologies brought us a leap forward in dealing with these challenges; however, they are predicted to be unable to cope with anticipated (nearly eightfold between 2015 and 2020 [1]) growth in traffic demands, driven by quick evolution of wireless devices in kinds and amounts. The picture is overshadowed by emerging huge amounts of various machine-type devices, taking the stage in the upcoming Internet of Things (IoT) era. These technology challenges emphasize the drive for research of novel approaches under the 5G network architecture initiatives.

1.2 Potential solutions

Fortunately, the evolution of Information Technology (IT) and communications is not all about challenges. Over the last decade the evolution of technology in various directions like cloud computing, virtualization, high speed fiber optics communications, efficient signal processing, advances in data center IT architectures, etc., has provided researchers and engineers with powerful tools for elaborating future-proof solutions to modern challenging demands.

Heterogeneous multi-radio architectures for mobile networks (HetNets) are modern solution for service providers, brought to address emerging connectivity demands by hierarchically adding smaller and smaller cells. The resulting setup allows user device to interact with the infrastructure via multiple radio access technologies (RATs). While more and more user devices are being equipped with multiple radio transceivers, this architecture allows mobile network operators to significantly improve network capacity by efficiently utilizing spectra of these radio technologies, thus offering higher quality of experience to their customers.

Thorough study of integration of new RATs, and understanding of required intelligence sharing between user equipment (UE) and infrastructure, for efficient use of these technologies, are envisioned to be fundamental enablers for future 5G standards. A confirmed example of benefits brought by HetNet architectures could be performance improvements in an unlicensed-band network, like WiFi, by leveraging centralized control from designated entity in mobile network core, like 3GPP LTE [2].

The agility and efficiency of modern software development processes have made

software elaboration, operation, and evaluation principles very attractive to other technology fields. Software Defined Networking (SDN) paradigm comprises a set of automation and abstraction concepts aimed to bring networking closer to its ultimate goal of interconnecting users and applications in the most efficient way, by providing entities that use the network with tools to shape network services to their needs. Within the scope of this work, SDN vision aligns well with HetNet architectures in desire to optimize resource allocation through centralized decisions, based on end-to-end view of traffic flows, and in attempt to involve both user and network sides of communication into optimization process, while allowing UE to efficiently use all the available communication technologies. While, the term SDN is relatively new, and there is a lot of debate around SDN concepts going among researchers, engineers, and equipment vendors, the principles of network control automation, abstraction from underlying network functionality, and software based flow switching laid foundation for prototyping within this research work.

Another networking solution proved useful in research and development of new architectures is the concept of *overlay* networks. Building another logical network layer on top of existing transport network is useful in highly dynamic environments like data centers, or in cases where major changes to *underlay* transport network are very expensive, like in large scale operator networks. Within this work, configuring tunnels over different provider networks and different technologies allowed to achieve desired connectivity without modifying underlying network protocols, which otherwise would require open access to a live cellular installation. The drawback is increased overhead in the network, but it does not interfere with research objectives and is acceptable for the scope of this work.

1.3 Overview of tools

Open VPN

Open VPN is a network tunneling software. Open VPN is widespread due to its openness, and availability for a variety of platforms. Particularly interesting for this project is its capability to dynamically deploy and execute custom applications or scripts on connecting client, based on triggered events, and capability to run as remote gateway interconnecting multiple private networks.

Open vSwitch

Open vSwitch is a multilayer software switch. Important feature for the project is its ability to expose forwarding functions to remote entities for programmatic extension and control via protocols like OpenFlow.

VMware vSphere

Easy to use and well documented virtualization solution. Used to abstract and share physical server resources to multiple virtual machines. Important feature is its robustness.

Docker

Software packaging solution, allowing to isolate an application at OS kernel level together with all required dependencies, files and network interfaces in a standardized unit – container. Used for persistent and automated deployment and runtime across various environments.

GRE

General Routing Encapsulation is simple tunneling protocol used to build overlay networks on top of IP networks. Important feature is its capability to encapsulate various protocols including Ethernet.

NFV

Network Functions Virtualization is a concept complementary to SDN. NFV suggests new way to implement and operate network functions, by abstracting them from hardware appliances and using virtualization techniques to package these functions as virtual machines running on generic hardware.

1.4 Structure of this work

This work is concerned with prototyping solutions to address several challenges in streamlined delivery of connectivity with high service experience over the multi-radio heterogeneous deployments to a variety of modern handheld and wearable mobile devices. Chapters 2 and 3 cover the author's work within the research on enabling technologies, namely Heterogeneous Networks [2, 3] and Device-to-Device communications [4, 5]. Chapter 4 describes research on prospective applications based on these enabling technologies [6, 7]. This thesis is structured as follows.

- Chapter 1 introduces evolution trends in modern wireless networks. Driven by all increasing connectivity demands, research and engineering communities are working on defining and implementing next generation networks. Main motivators for the research are mentioned in Section 1.1, and the following Section 1.2 outlines research areas of the group within which this work was done.

- Chapter 2 focuses on prototypes and demonstrators developed to support research done on optimization problems in cooperative radio resource management in Heterogeneous Cloud Radio Access Network (H-CRAN). Section 2.1 describes coordination issues in prospective 5G deployments, and introduces a combination of HetNet principles with C-RAN. Section 2.2 outlines coordination schemes evolution, concluding with envisioned H-CRAN architecture. General features of the testbed implementation are described in Section 2.3 with technical details specified in Section 2.4. Chapter is concluded with the list of achieved results presented in Section 2.5.

- Research activities outlined in Chapter 3 aim to use traffic control principles similar to Chapter 2, by utilizing direct link between proximate users to offload communications from infrastructure. Section 3.1 begins with introduction to main factors driving the search for new ways to increase network capacity. Section 3.2 introduces research on improvements in D2D communications, brought by network assistance. Solution prototype is described in Section 3.3, with technical details laid out in Section 3.4. The results of prototype implementation process are presented in Section 3.5

- Chapter 4 elaborates on security and trust research in heterogeneous wireless ecosystems. Section 4.1 outlines research conducted on a novel algorithm

to maintain security functions of proximate devices in case of unreliable cellular connectivity, covering cases when a new device joins the secure group of users or an existing device leaves it. Section 4.2 addresses prototyping to support study of applicability of modern cryptographic primitives, including the pairing-based cryptography, to another emerging area where security is vital – new kind of smart electronics known as *wearables*. Section 4.3 suggests potential services enabled by an integrated heterogeneous wireless ecosystem.

- Chapter 5 concludes this work with an assessment of implemented solutions.

2. CENTRALIZED RADIO RESOURCE MANAGEMENT IN HETNETS

5G communications ecosystem targets to reach higher rates never challenged before, and to keep up with these rates, modern heterogenous network architectures have already developed improved ways to integrate various RATs. Seeking new vectors for evolution, emerging paradigm of heterogeneous cloud radio access network (H-CRAN) merges RAT integration with advanced cloud infrastructures. This approach enables improved management on the network-wide scale, allowing to implement cross-cell radio resource allocation in a coordinated way. Recent research addressed the gaps in theoretical performance analysis, and provided assessment and mathematical methodology for *real-time* optimization of cooperative radio resource management in H-CRAN [2]. Resulting algorithms allow to balance between throughput and fairness metrics in a flexible way, as might align with network operator's development plans. Also, this approach demonstrated some advantages over state-of-the-art multi-radio resource allocation schemes.

This chapter introduces work on elaborating practical implementation of a proof-of-concept prototype, to demonstrate feasibility of algorithms for cooperative resource management in H-CRAN.

2.1 Introduction and motivation

In the upcoming 5G era, most of the modern conveniences could use wireless networks to push our understanding of quality of life even further, and provide even broader spectrum of services. This vision poses unprecedented challenges to those who research, develop, deploy and operate theses networks as the demand for mobile data traffic is expected to increase nearly eightfold in the period from years 2015 to 2020, reaching as much as 30.6 exabytes globally per month [1]. To provide service matching these demands, wireless networks would be required to support very dense

user device and network infrastructure nodes placements, employ very high carrier frequencies including emerging millimeter wave (mmWave) technologies, and utilize larger numbers of antennas in massive multiple-input multiple-output (MIMO) installations [8]. Together with harnessing additional spectral resources, network management needs to be improved, to be able to operate prospective 5G deployments in an intelligent and flexible way, also paying attention to increasing importance of power and cost effectiveness. An emerging paradigm of HetNet is expected to be the advanced networking architecture to address these challenges. Utilizing 3GPP LTE macro cells for complete coverage, this architecture is enhancing connectivity and capacity, by augmenting infrastructure with small cells varying in RAT and in size. This approach allows to use both licensed and unlicensed spectra, supporting user access in open, closed or hybrid fashion [9], and to connect users through technologies like pico and femto cells, remote radio heads (RRHs), relay nodes, WiFi and WiGig access points, integrated WiFi-LTE small cells, etc. Significant gains in capacity and overall user experience can be achieved by applying efficient coordination to this variety of RANs [10]. However, to maximize the flexibility of HetNet management for robust interference mitigation, mobile connectivity continuity, and enhanced capacity [11], all of the macro nodes and small cells have to be interconnected via low-latency high-rate backhaul links, which significantly increases capital expenditure (CAPEX) and operational expenditure (OPEX).

Different coordination schemes can be applied to 5G HetNet architecture depending on backhaul available to network operator. In case backhaul deployment is restricted to *non-ideal* infrastructure, the coordination can be performed in anchor-booster architecture, where general network management is provided by anchor macro base station (BS), and multi-radio small cells offer opportunistic traffic offloading capability to boost user data rates. Otherwise, if backhaul is providing higher capacity and lower latency, by using *near-ideal* carrier (e.g., optical fiber), low-power small cells can offload processing of baseband signals, by forwarding them to a remote centralized server platform. Network operators with wide fiber infrastructure deployments tend to prefer the latter approach, named Cloud RAN (C-RAN), by employing inexpensive wireless installations for front-haul connections, primarily in areas with high traffic demands requiring ultra-dense HetNet deployments. This way the concept of C-RAN allows to substantially lower operator's CAPEX and OPEX, considering that the largest part of infrastructure investments by a network operator are spent on RAN part [12]. Additionally, C-RAN architecture is much more efficient in terms of wireless infrastructure energy consumption.

The RRH unit in C-RAN is a simplified low-power node, acting as a soft relay by compressing baseband signals from mobile UE and sending them to the centralized base band unit (BBU) over high-rate front-haul links. In such setup, front-haul connections can become capacity bottlenecks, thus requiring advanced signal processing solutions [13] and dynamic resource management [14] to maintain needed levels of performance. Some key points shaping C-RAN capacity include

- Proper optimization [15] addressing practical backhaul constrants [16]
- Uplink RRH association strategies [17] with corresponding restrictions on implementation complexity and radio resource consumption [18]
- Employed decentralized beamforming algorithms [19] and large-scale distributed MIMO-aware power and antenna selection schemes [20]

Combination of HetNet principles with C-RAN "signal processing cloud" has recently emerged as the Heterogeneous Cloud RAN (H-CRAN) concept, aimed to further improve cooperative gains in a cost-efficient way. H-CRAN takes best from both worlds, by blending networking techniques of heterogeneous networking with cooperative processing of cloud computing, thus enabling facilitated interference mitigation, scalable deployments, and efficient radio resource control. The function of managing radio resources of low-power nodes (LPNs) is delegated to a virtual BS, which runs in the cloud and is allocated from total processing capacity available at the physical BBU pool. Currently, main research focus was to outline technological features and base principles of H-CRAN to lay foundation for commercial H-CRAN based 5G systems [21]; however, to gain ultimate understanding of its capabilities there are still major challenges in the fields of theoretical performance analysis and optimal resource allocation to be addressed.

Recent research has focused on the problem of cooperative radio resource management in 5G-grade H-CRAN systems and provided a comprehensive methodology for real-time performance optimization of H-CRANs [2]. Proposed solution allows to dynamically control the amount of resources allocated to end users, for two alternative metrics of interest, namely, the fairness of resulting resource shares across all the available RANs, and the overall system throughput. This work describes implementation of several testbeds, to support the research on exploiting trade-offs between fairness and throughput metrics, and development of proof-of-concept

demonstrations of network-centric, network-assisted, and UE-centric resource allocation mechanisms in characteristic H-CRAN environment, with different levels of available LTE/WiFi integration.

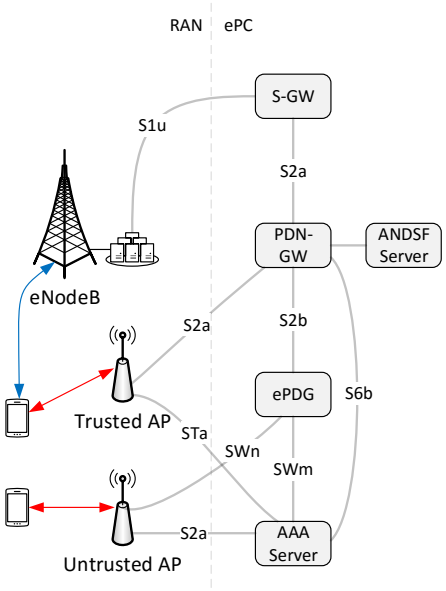
2.2 H-CRAN: technology and standards background

Although H-CRAN technology is still not ready for commercial use in production networks, coordinated use of multiple RATs within single multi-radio network managed by the operator is becoming actively researched in 3GPP. This section reviews some of the respective efforts.

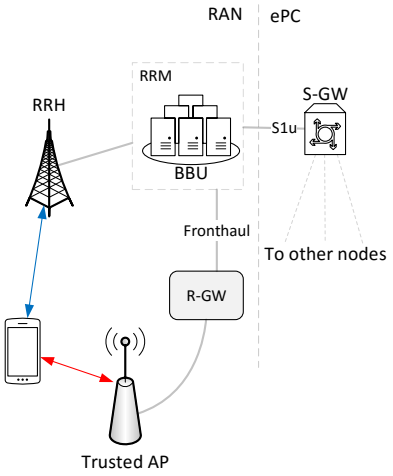
3GPP Release 11 introduced loose coordination model, where access network discovery and selection function (ANDSF) is managing interworking between WLAN and 3GPP technologies (see TS 23.402), by means of ANDSF policy server inside core network. The operator specifies discovery and WLAN resource usage within the network, by defining relatively static policies, and the UE makes actual network selection based on local operating environment and changes in radio link conditions. ANDSF-enabled architecture is shown in Figure 2.1(a).

Major drawback of ANDSF is that *UE-centric* decisions are sub-optimal as UE does not have complete vision of other users sharing radio network, their link conditions and radio resource requirements. Also, multi-radio cells within HetNet deployments do not share information about each other's state, conditions, and radio resource usage. This diminishes network-wide radio resource utilization even further. Additionally, placement of mobility anchor in the network core (at P-GW typically) makes it very expensive resource-wise to allow UE to steer traffic across WLAN and 3GPP links when link conditions change rapidly.

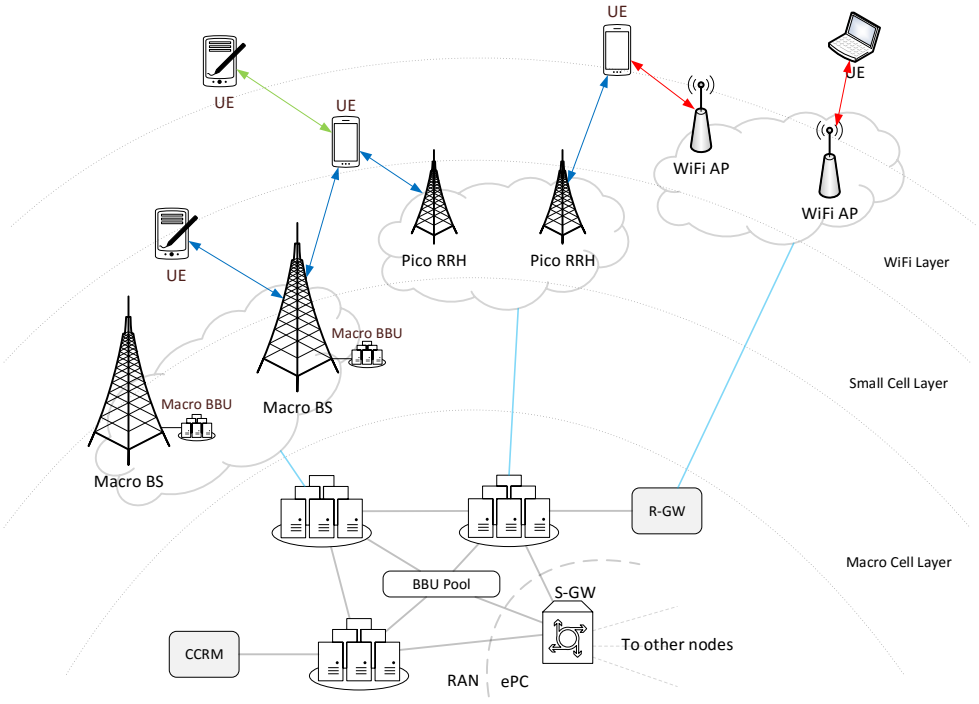
These WLAN/3GPP radio interworking issues were alleviated to some extent in Release 12 (see TR 37.834). Traffic steering between WLAN and 3GPP RANs can be facilitated by setting link quality and WLAN load thresholds at cellular BS (eNB or eNodeB) through internal coordination within RAN. However, the standards did not specify coordination required in the network to set appropriate thresholds. Also, steering traffic in a flexible and efficient way is still not feasible as mobility anchor of WLAN/3GPP links is still defined in the core network by the standard. The mechanisms based on this integration option will be referred to as *network-assisted* in this chapter.



(a) ANDSF-enabled architecture



(b) Integrated anchor-booster system design



(c) Centralized Management in H-CRAN

Figure 2.1 *Example of H-CRAN deployment and system modes*

To make integration of WLAN within 3GPP RAN tighter, recent proposals (see e.g., RP-140685, RP-140738) are targeting to incorporate WLAN as a secondary carrier within the 3GPP RAN, anchored at the eNB. Release 13 is expected to consider the proposed architecture for standardization, extending benefits of system design introduced for 3GPP small cells by Release 12 dual connectivity anchor-booster system design (see TR 36.842), and extending existing 3GPP carrier aggregation framework to also include non-3GPP RATs e.g., WLAN, like in the illustration used in this chapter.

If 3GPP will also consider standardizing the interface between eNB and WLAN access points (APs) for non-collocated WLAN/3GPP deployments, this integrated *network-controlled* architecture will enable coordinated radio resource management for non-3GPP WLAN networks thus extending the benefits of LTE-based anchor-booster schemes. Also, anchoring WLAN connections at eNB will allow users to delegate WLAN control and management functions to LTE network and use WLAN capacity solely for data offloading. Figure 2.1(b), shows simplified architecture of this approach. Radio resource management function is marked as RRM module in the plot, and designated gateway performing interface matching to connect WiFi and BBU is marked as RAN gateway or R-GW in the figure, as respective interface is not yet standardized.

Another 3GPP study (see TR 37.870) is currently exploring architecture for multi-RAT networks with enabled radio resource coordination across anchor cells, in addition to coordinated use of radio resources within anchor cell coverage area, thus improving overall system performance even further. eNB to eNB coordination in the distributed model may be achieved over X2 interface; however, in practice, different approaches may be used e.g., utilizing a centralized radio resource controller to manage radio resources system-wide.

Cloud RAN architecture becomes feasible in deployments, where high-rate fiber connections are available in the backhaul. Linking simple RRH with restricted functionality to centralized BBU pool within the cloud, such architecture wraps entire RAN functionality within a single centralized node, thus significantly facilitating introduction of non-3GPP RRH nodes to allow for centralized multi-radio coordination. However, this requires additional standardization efforts from 3GPP.

Using principles similar to the multi-radio Cloud RAN, conceptual H-CRAN architecture introduced in [21] and reviewed in Section 2.1 enables centralized processing

within the EPC, by connecting nodes to centralized server using S1 interface. Resulting centralized network control allows to perform coordinated/cooperative radio resource management, basing decisions on additional factors like variations in load (e.g., busy hour effect) and UE mobility (e.g., by offloading high mobility UEs to macro cell by default).

This chapter focuses on a H-CRAN deployment, where centralized management of system radio resources is performed by a dedicated entity, named Cooperative Radio Resource Manager (CRRM), and assuming same X2 backhaul interfaces for connection to the CRRM server. Deployment and system model of this approach are shown in Figure 2.1(c).

2.3 Practical testbed implementations

The following step of this research is to implement considered algorithms in an operational testbed, to study their practical performance. This section discusses realistic requirements for the implementation of proposed methodology. For this purpose, it is required to integrate resource allocation algorithms into the protocol stack of modern LTE and WiFi networks.

2.3.1 Implemented R&S demonstrator

Recently, our research group at TUT, including author of this work, has completed underlying HetNet testbed configuration, based on LTE eNodeB emulator, CMW500, by Rohde and Schwarz[1]. In this experiment, CMW500 was readily set up for a scenario of offloading traffic from LTE to WiFi, to assess capabilities of CMW500 equipment for 5G research. Integration example demonstrated a simple handover scheme for WiFi-LTE data offloading case, where WiFi signal strength was main deciding factor. CMW500 was configured with LTE FDD cell signaling on frequency band 7, downlink 2645 MHz, uplink 2525 MHz, total bandwidth = 20 MHz. WiFi AP and CMW500 were bridged at the same server, which was also configured to lease IP addresses to UE via DHCP. An RF Shield Box model CMW-Z10 was used to control WiFi link quality. The shield box was connected to CMW500 unit via RF1 COM port and served as antenna for cellular link. This way, UE

[1]Video presentation of the experiment with R&S CMW500: http://winter-group.net/rohde-schwarz-tutorial/

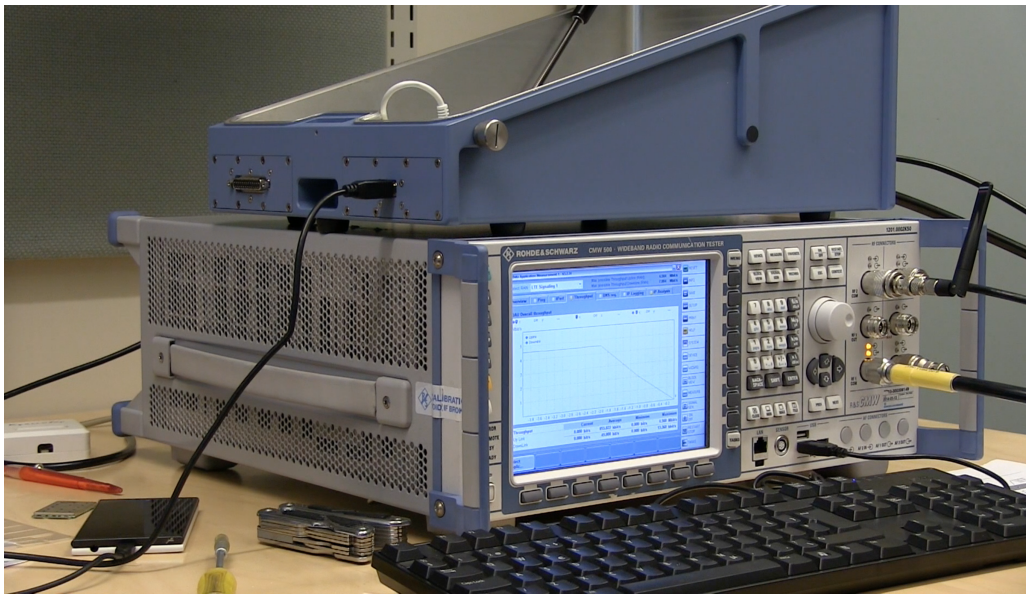


Figure 2.2 *Setup with Rohde and Schwartz CMW500*

placed into open shield box connected to gateway server via both LTE and WiFi and was able to use the link with a better quality. Closing shield box lid introduced over 80 dB of digital attenuation for WiFi, thus triggering UE to switch its data transmission to LTE. When the lid was open again, WiFi link quality became better than that of LTE, and traffic was offloaded back to WiFi. CMW500 capabilities to emulate LTE provider's network allowed to quickly setup a testbed for improved offloading logic design, for the purposes of 5G research. Further research targeted intelligent data offloading, by employing advanced LTE link parameters. Figure 2.2 demonstrates a short video presenting the setup. The video is available at http://winter-group.net/rohde-schwarz-tutorial/.

2.3.2 H-CRAN architecture prototype

Current prototype implements a multi-radio scenario, where a cellular network is coupled with a WiFi access point, thus providing UE with a possibility to seamlessly switch between two RATs, or to efficiently use both of them at the same time. Naturally, implementation of the proposed model requires access to cellular BS side. It also requires UE capable of exposing necessary control information over the available radio channels. However, development kits for mobile platforms currently available on the market provide very limited support to manipulate corresponding interfaces and data flows.

In this implementation, the UE was connected to a conventional cellular network and a separate local WiFi AP, both providing Internet connectivity via different ISPs. To simulate a common network behind both radio links, mobile phone establishes two VPN connections bound to cellular and WiFi interfaces, respectively, and terminating at an aggregator node (which may be e.g., located in the Internet). The aggregator node, bridging two links, simulates a packet gateway in LTE network.

To aggregate both links on UE side, Open vSwitch (OVS)[2] was employed. Correspondingly, vSwitch daemon, running on the phone, includes links representing WiFi and cellular connections together with a local virtual interface, serving as a bind point for outgoing traffic, generated by applications using *sockets* API. However, specification of OVS assumes that all links in the virtual switch are able to handle Ethernet headers, which is not the case for cellular network exposed to system as a point-to-point RmNet interface. However, this issue could be solved by augmenting vSwitch to allow this type of interfaces, or by utilizing a module that would skip Ethernet header processing by an offset. In this testbed, author chose to add another layer of tunneling on top of existing VPN with a Generic Routing Encapsulation (GRE) tap tunnel (also referred as Ethernet GRE or EGRE) as the simplest in implementation, and for consistency, GRE tap layer has been added to WiFi VPN as well. Corresponding protocol stack of UE is outlined in Figure 2.3. Similar architecture is used at the aggregator on simulated operator side. This way, on the top level of abstraction architecture folds into two switches interconnected with two redundant links. Overall system topology is illustrated in Figure 2.4.

The proposed setup allows us to implement a controller that would be able to maintain vSwitch's forwarding table using OpenFlow protocol[3], and provide desired levels of per-flow traffic steering (i.e., resource allocation) based on dynamic criteria. A separate module collects data from RATs about the state of every underlying physical link, such as Received Signal Strength Indicator (RSSI) and radio signal strength. The controller uses these data to efficiently assign new flows to either of outgoing interfaces. Coming back to the analytical algorithm implementation, this means that it is possible to transfer implemented resource allocation logic (e.g., relative fairness scheme) into the OpenFlow controller and specify control channel interface (which is LTE in this case).

[2]Open vSwitch website: http://openvswitch.org/
[3]OpenFlow website: https://www.opennetworking.org/

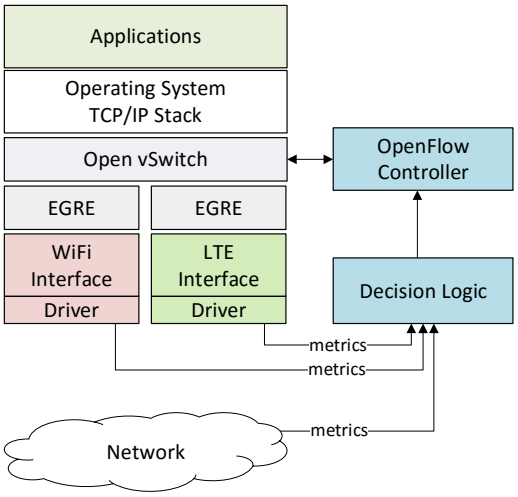


Figure 2.3 *Proposed testbed topology for multi-RAT HetNets*

2.3.3 Infrastructure for the prototype

The network side entities were installed as virtual machines (VMs) running on virtualization platform, built with vSphere[4] from VMWare. This approach allowed to deploy and interconnect architecture elements in a highly flexible and dynamic fashion. Virtualization platform was designed and assembled by a smaller group of research assistants, including the author of this work, to support various research activities performed by the larger group.

For the research scenario, WiFi access point was connected to the Internet through university network and configured as a router between wireless and wired sides. UE was connecting to both WiFi and LTE networks and had Internet connectivity through either network. Using specific static routes the UE was contacting particular VPN gateways over certain technology i.e., IP address of the gateway terminating LTE side VPN was routed via LTE network, with similar behavior for WiFi side. On the network side, traffic to the VPN aggregator node was chained by the platform through an entry node switch that was splitting traffic destined to LTE VPN and WiFi VPN, and then passed through WAN link emulator node. Details of such service chaining are illustrated in subsection 2.4.3.

[4]VMWare vSphere solution: https://www.vmware.com/products/vsphere

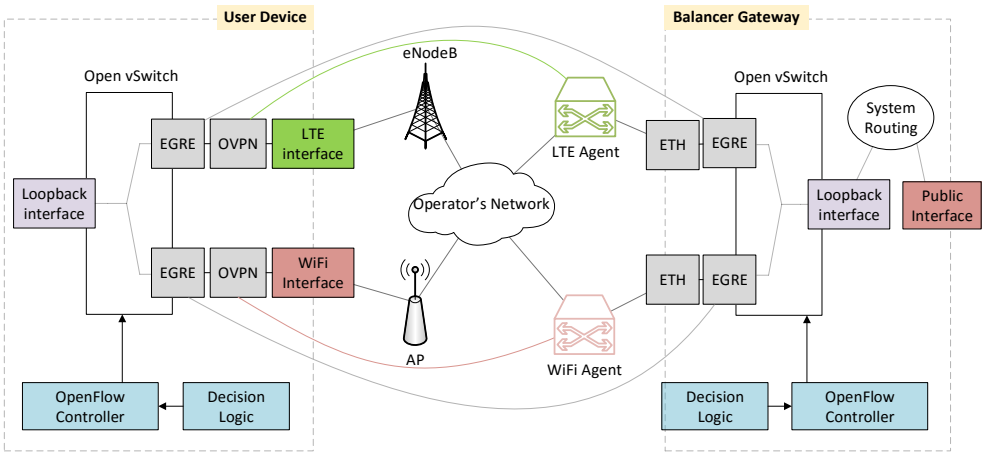


Figure 2.4 *Overall architecture of prototype implementation*

2.4 Technical details

Most important features of current prototype implementation are: software based switching, network control automation and *overlay* network over regular provider networks. Flow based switching was implemented with Open vSwitch software and network control was performed with OpenFlow controllers, both described in subsection 2.4.2. Design of VPN and processes of measuring and influencing link conditions are outlined in subsection 2.4.3. Significant work was done to prepare UE to handle these tools, subsection 2.4.1 covers modifications to the UE and procedures to enable it for research scenarios. Finally, architecture is summarized with the workflow of a research scenario in subsection 2.4.4.

2.4.1 Toolchains and custom software on user equipment

A suitable UE platform for testbed implementation has been the SailfishOS[5] running on Jolla phones[6], which combines *Linux* and *MER* software[7], and provides a flexible platform software development kit (SDK). The platform offers all required tools to build custom kernel modules, as well as generic GNU\Linux software. Hence, installing OVS, GRE, and Virtual Ethernet (VETH) modules, as well as OVS database and userspace tools, has been fairly straightforward. The software in SailfishOS is packaged with RPM Package Manager (RPM) (originally Red Hat Package Manager,

[5]Sailfish OS website: https://sailfishos.org/
[6]Jolla phone website: http://jolla.com/
[7]MER project website: http://merproject.org/

now a recursive acronym), which provides powerful tools to integrate third-party software and to install additional components.

Some of the tools used in prototype require integration at OS kernel level, which means that UE kernel has to be built with corresponding functionality enabled. Customization of MER based Sailfish OS is done through Platform SDKs provided by the project. These platform SDK tools include compilers, Scratchbox2 cross-compilation toolkit, MIC image creator, Zypper package manager and other instruments to make development easier. In this setup, SDKs are run as an image in a *chroot* environment; however, SDK can also be used as a dedicated virtual machine. SDK was installed on development machine from a *rootfs tarball* containing essential tools for MER platform development. The goal of using platform tools was to compile GRE, Open vSwtich, and VETH kernel modules for SailfishOS running on UE. As a basis for the new kernel configuration, author extracted configuration file from running system of a Jolla device. Configuration file was edited to include appropriate modules and was used for the build process, resulting in corresponding kernel object (.ko) files compiled. These module files were transferred to the UE and loaded into running kernel as part of prototype implementation. Another feature of the Platform SDK that was useful for us, is that provided *cross-compiler toolchain* can be used to build custom userspace software, which is otherwise not available in official repositories. Using this *toolchain*, author was able to compile userspace part of OVS software.

2.4.2 OpenFlow software switch

OVS is a multilayer virtual switch, designed to target at multi-server virtualization deployments. It provides a lot of important features for highly dynamic environments at large scales; however, for this prototype the most important feature of OVS is that it adheres to the emerging Software Defined Networking (SDN) paradigm and thus supports forwarding based on flow tables and can use OpenFlow as a method of exporting remote access to control traffic forwarding. These features enable flexible placement of traffic control entity (at UE itself, on network side, or combination of both), and per flow control over traffic forwarding, meaning that certain sessions can be selectively placed on WiFi or LTE interfaces in a dynamic way. OVS consists of kernel side module, responsible for actual packet forwarding in the *data-path*, and userspace daemon, interacting with network state database and exposing control functions to external entities.

OpenFlow based SDN vision implies that forwarding tables of a device should be maintained by a separate entity – the controller. This controller is responsible for making forwarding decisions and programming them into the flow tables along the data path, using Open Flow protocol. The controller in this installation was based on POX controller – a networking software platform written in Python [8]. Controller assumes the management of UE's forwarding plane consisting of 3 interfaces – WiFi interface, LTE interface, and internal system interface. Main task for controller is to run one of the optimizations algorithms supplied by researchers. Algorithm implementation receives metrics feed from the software, performing measurements of network conditions on both radio interfaces, and reallocates incoming and outgoing flows accordingly.

2.4.3 Network side

In the absence of access to an operational cellular network installation, researchers have to simulate joint control of WiFi and LTE networks for UE using VPN tunnels. This testbed used Open VPN for its simplicity and wide platform adoption. Clients would connect to dedicated VPN anchors through WiFi and LTE links, this way allowing the laboratory network to control setup of tunnel interfaces in a similar way as real operator network would control setup of physical links.

The authentication in VPN network is based on Rivest-Shamir-Adleman (RSA) certificates, as opposed to pre-shared key based authentication. Using EasyRSA tools, author has created certificate hierarchy where every user device would receive its own certificate to authenticate in the VPN network, and, based on information in that certificate, network would uniquely identify the user and configure it with predefined features.

Network-wise, VPN anchor was configured for *subnet* layout. Unlike in *point-to-point* layout, where a dedicated subnet is allocated for every client-server link, in *subnet* layout all clients share common address space which slightly facilitates routing on the server side for prototype purposes.

During start up process, VPN agent places policy based routing (PBR) rules into routing subsystem, to selectively route traffic from VPN network to the gateway facing the Internet to provide external connectivity. Configuration of Open VPN

[8]POX controller: https://openflow.stanford.edu/display/ONL/POX+Wiki

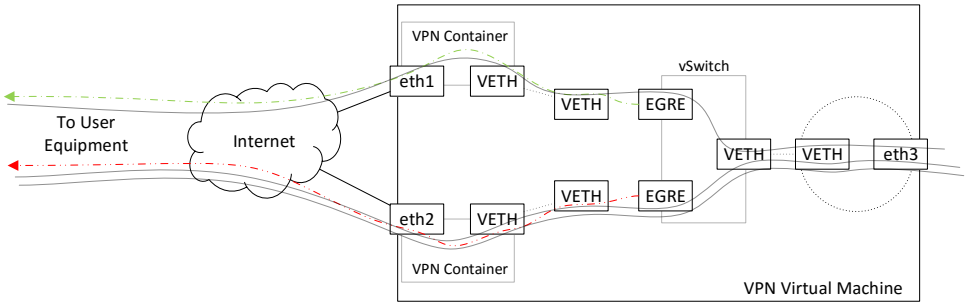


Figure 2.5 *Detailed network architecture of VPN server*

server is provided with example of LTE agent in Listing 1

To separate operation domains of WiFi and LTE side VPN servers, without the need to run them on different virtual or physical nodes, both processes are executed on the same VM, isolated in dedicated Docker[9] containers. Docker is a set of tools that provide convenient way to manage Linux container (LXC) environments. Every container running *openvpn* process has in its namespace a dedicated host network interface, facing the client, and a VETH link bridged with host network. This way, *openvpn* container acts as a router between host side network, leading to internal operator network, and client network behind VPN network. Detailed network layout of the VPN server is presented in Figure 2.5. Container mounts into its internal filesystem a directory with corresponding configuration files and certificates (e.g., LTE directory for container running LTE side) and executes *openvpn* process according to these configuration files.

Most of commercial off the shelf (COTS) WiFi access points are shipped with proprietary, closed source operating systems, and are built with network chips run by closed drivers from hardware vendors. This way, access point becomes a "black box", exposing enough control to set up the network, but prohibiting any new functionality on top of the hardware. For an access point to be useful in such research, both operating system and wireless interface firmware must be open, to allow modifications and give access to internal variables. A good example of such device is a dual band WiFi router from Linksys model WRT1900AC. For this research the device was installed with OpenWRT[10] system, custom compiled to include WiFi drivers as modules, rather than built into the kernel, making it easier to dynamically add

[9]Docker platform: https://docs.docker.com
[10]OpenWRT system: https://wiki.openwrt.org/doc/start

changes into driver's code. One of the most important changes to the operating system was to enable Linux *debugfs* filesystem that exposes multiple system values, like status of numerous transmit queues and current transmission rates.

This way, AP was prepared to send measurements of interest to the process implementing forwarding optimization logic, where the latter would combine WiFi state data with LTE data, or any other information found useful by researchers, like prioritization of users and RANs, and would yield new forwarding rules for UE side and operator side.

To simulate various network conditions, all traffic to the emulated aggregator node on network side was routed through a dedicated VM running WANem – a Wide Area Network Emulator[11]. This allowed us to impose various WAN characteristics common to LTE networks, like network delay, packet loss, packet re-ordering, jitter, etc. to the links that are in fact running in research laboratory LAN. Such service chaining of traffic dedicated to one VM through another VM is not natively supported by basic feature set of available vSphere installation, so several safety policies had to be disabled, and in such cases certain broadcast and unknown traffic should be suppressed by virtual switch, and greater caution should be paid when routing traffic, to avoid forwarding loops. Network scheme of traffic flows in the setup with WAN emulator node is demonstrated in Figure 2.6.

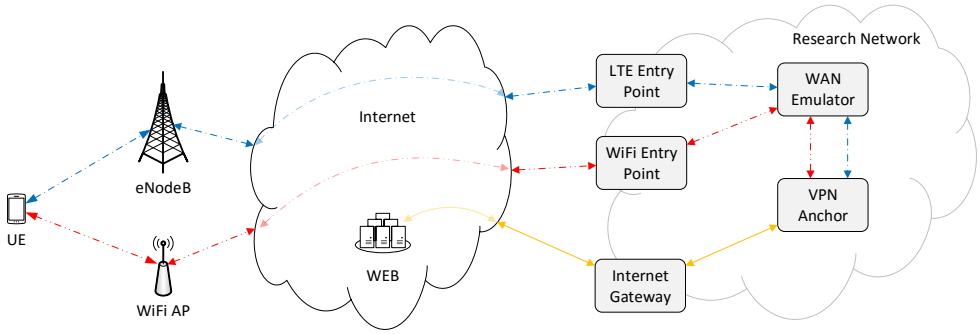


Figure 2.6 *Setup with WAN emulation node*

2.4.4 Architecture summary and workflow

Network side of the deployment consists of a number of VMs serving as VPN anchor, WAN Emulation, and Internet Gateway. When VPN anchor VM boots up, it

[11]WANEM emulator: http://wanem.sourceforge.net

performs the following steps

- starts VPN processes isolated in containers
- moves dedicated network interfaces to corresponding container namespaces
- creates a VETH pair for each container
- moves one end of every VETH pair to corresponding container namespace
- runs GRE tap tunnel to client over second end of VETH pair when client connects
- adds GRE tap interface to OVS
- creates another VETH pair for host networking
- adds one end of host VETH pair to vSwitch
- sets second end for routing to internal operator network

Code listing of the script starting the service is provided in Listing 3

When UE side and network side virtual forwarding planes are ready, both devices start measuring link qualities. Depending on coordination scheme scenario, controller software is started on either side and both virtual switches connect to it. Measurements from UE, AP, and operator side are also sent to the controller. This brings the setup to final operational state, where controller would use all available metrics, to run resource optimization algorithm provided by researchers, and would re-allocate traffic sessions across available links accordingly.

2.5 Conclusions on coordinated resource management

This chapter considered implementation of testbeds and demonstrators for H-CRAN – a recently emerged concept, brought to improve available cooperative gains in HetNets in a cost-efficient way. Elaborated solutions focused on improving research of coordinated radio resource management problem in 5G-grade H-CRANs, by implementing a platform to analyze their real-time performance optimization. The most important findings from development process are summarized below.

- Implementing a prototype for the CRRM unit helped to outline envisioned H-CRAN system architecture, based on review of respective 3GPP standardization processes and technology implementation options.

- Proposed architecture demonstrates viability of virtual forwarding planes concept for prototyping in traffic optimization research. Building an *overlay* network allows to abstract from actual network technologies that are closed to modifications, while preserving physical channel properties.

- Decoupling of measurement functions, traffic forwarding, and control enabled dynamic deployment of user centric, network assisted, and network controlled research scenarios, or other hybrid control schemes. Also, coordination logic can use any other information sources e.g., an even broader view on network at Internet scale from operator's border gateway protocol (BGP).

- Using modern virtualization tools has proven to be crucial for fast and agile prototyping of new network architectures. This approach to packaging of software components allows to easily reproduce testbed setup in any hardware or virtual environment, in an automated way. Virtualizing architecture entities also facilitates scaling up to allow wider setup, as well as scaling down to run self-contained installation on a researcher's laptop.

- Resulting modular testbed allows to deploy new steering algorithms in a quick and automatic ways. The setup offers researchers an environment for testing traffic flow optimizations, similar to what the new architecture would be like in a live network.

- Prototype implementation improved research on H-CRAN resource optimization with CRRM published in [2] and study on Prioritized Centrally-Controlled Resource Allocation in Integrated Multi-RAT HetNets appears in [3]

3. NETWORK ASSISTED DEVICE-TO-DEVICE COMMUNICATIONS

The offloading principles, discussed in Chapter 2, are not limited to be used only in network operator's infrastructure. Similar techniques could be used to improve network performance, by offloading communications between proximate users from infrastructure to direct link between users, or device-to-device (D2D) links. Next sections cover research and prototyping activities performed in exploring improvements to user traffic offloading from infrastructure network onto direct D2D links brought by assistance from cellular network.

3.1 Introduction and motivation

The increase in mobile data traffic pushes network operators to seek ways to relieve congestion in their infrastructures. A natural way to mitigate the shortage of available radio resources is to deploy an increasing number of various sized BSs; however, such approach is costly and faces some practical challenges. An alternative approach would be to leverage the capability of modern UE to establish simultaneous connections via different radio links, and to enable traffic offloading from cellular network onto D2D connections in unlicensed bands like WiFi. The problem with WiFi is that there is no fast and efficient method of service or device discovery built into the protocol, as well it lacks functionality to efficiently manage multiple D2D links. Recent research [5] demonstrates that these limitations can be solved by a certain amount of network assistance to D2D communications. This part describes network-assisted D2D technology prototype developed by research group at Tampere University of Technology. The resulting solution is completely standards-compliant and provides service subscribers with seamless D2D connectivity. The link layer technology for D2D connections in the prototype is WiFi Direct (WFD); however, described techniques can be applied to other potential D2D technologies. Neighboring devices could communicate over D2D channels whenever possible to

reduce network operator's reliance on smaller-scale and denser cell infrastructure. The Third Generation Partnership Project (3GPP) has put a lot of effort to define a licensed band D2D technology [22]. Also, there are proprietary solutions being developed by individual companies [23, 24]. However, standardization process takes time, as well as adoption of standards by device manufactures. But when looking towards the unlicensed bands, there are several D2D technologies already available in most client devices. However, there are several engineering limitations to those technologies, with primary one being lack of efficient connectivity management and device discovery. Considering that battery life is a scarce resource for most users, the amount of energy spent on device discovery and D2D connection negotiation is unacceptable. The group proposed a scheme of network management for unlicensed-band D2D connections with the example of WiFi, where users will benefit from their cellular network connectivity to help manage their D2D connections. The boost in cellular network capacity after introducing network-assisted D2D was demonstrated in [25, 26]. Standardized solution for network assistance is expected to be decoupled from actual D2D link layer technology. An example of a technology suitable for D2D is WFD [27]. WFD is already available on the market and offers high data rates over mid-range point-to-point (P2P) connections. Additionally, WiFi is already considered for cellular traffic offloading [28], and thus this work primarily focuses on using WFD as offloading technology. Primarily, the goals of this research were to elaborate a 3GPP compatible architecture to enable network-assisted D2D offloading, to make resulting solution compliant with current web standards, and to support suggested architecture with a proof-of-concept (PoC) implementation in real-world conditions, with offloading of cellular traffic (e.g., LTE) onto WFD links.

3.2 Network assistance in D2D communications

The benefits of short-range communications enlist higher data rates, lower transfer delays, and better power efficiency [29], as well as improvements in spatial reuse. Network providers are already leveraging these improvements, by introducing increasing number of pico- and femto- cells into their deployment. To gain even higher capacity improvements, natural next step on the way to the vision of 1000x increase [30] in upcoming 5G systems, by year 2020, for providers could be improving spatial reuse by enabling direct communication between clients. Most of modern COTS UEs are already supplied with built-in capabilities to establish direct links, but in its current state the technology is not suitable to be used in proximity services. Also,

providing seamless connectivity, when offloading cellular connections is challenging in current deployments (e.g., mobile IP). This work concentrates on seeking ways to resolve these issues in an efficient way, by adding network assistance to the system.

A lot of services nowadays can utilize proximity-oriented communications. This has attracted a lot of attention to research of D2D communications both from industry and from academia side. List of use cases that could potentially benefit from D2D connectivity includes context-aware applications, local voice calls offloading when users are in proximity, multimedia content streaming and sharing, gaming, and many others. As variance in D2D channel conditions can be significant in time, it is very difficult to offer certain service level agreement (SLA) with specific Quality of Service (QoS) on such links, thus offloading to D2D links has been considered mostly for delay-tolerant applications such as file transfers. However, in cases when users are mostly stationary and at reasonably close distance, offloading to D2D links becomes an attractive option for many other services like multicast video streaming, social gaming, etc.

Establishing D2D communication link has two basic requirements: a mechanism for devices to discover each other, and set of operations for connection setup. Current WiFi implementations fully support these features; however, not in the most efficient way, and this is where network assistance comes helpful. In case D2D link would fail, network assistance also ensures session continuity by restoring connectivity over infrastructure links. Suggested design introduces a new appliance that provides management of these assistance features from provider's network – the D2D server. This entity keeps track of subscribers UE together with any P2P application ID supplied by their users. Server is actively communicating with evolved Serving Mobile Location Centre (eSMLC) or any other entity providing positioning services in operator's core network; also D2D server has a control link established with UEs to assist with discovery, D2D connection setup and ensures selection of suitable communication link.

Additionally, D2D server is responsible for keeping up-to-date with user profiles and their D2D server names, by communicating with 3rd party application servers (also referred to as Content Registers, or P2P application servers, throughout this work). These content registers hold the links to P2P application IDs and their offered content or service published by network subscribers. Subscribers use P2P application servers to publish their content and search for other peers, who are

offering any potentially interesting content or service in proximity.

Figure 3.1 depicts the workflow in proposed architecture: (1) User authenticates with Content Register server in the usual way, by visiting to website or using a dedicated application, e.g. Facebook or any other social network could be used as P2P application server. Using an existing social network as content register allows all the publish/search operations with content to automatically obtain users' social context, e.g., share proximal content only with friends. (2) UE runs a software client, connecting to D2D server in its network, and authorizes it to update P2P application server with necessary information, to allow other users to request published content via appropriate D2D server. At this point, system ensures that UE system account indeed has access to the corresponding Content Register account. (3) Using functions provided by P2P application, the users can search for any other D2D ready content shared with them. Or they can publish new content, and post their intent to engage in proximal D2D connections with other users. (4) When one of the peers uses hyper-reference from Content Register, it is passed to D2D client software which subscribes UE to D2D server, and the latter resolves content location into actual D2D link-layer connection details. (5) D2D server keeps track of peers' location and instructs them to start using D2D link, when appropriate conditions are met. When P2P data exchange starts, Content Register is no longer involved. Communications channel is monitored by D2D server and adjusted accordingly to network state, as observed from the core and from UEs. As credentials and direct link parameters are derived by the network, devices themselves need no additional authorization or any direct contact prior to data transfer. Also, when transfer is complete and D2D link is dismantled, the devices are still unaware of each other's actual identity. This feature allows to further build anonymous sharing services, which are not possible in systems, where devices have to broadcast their presence to detect peers.

Suggested approach resulted in a system with several features:

- The system inherits community tested security and access models from P2P application server. D2D service users can shape access policies for their content to be visible only by certain peers. This implies that in order to be able to even locate the content, an eavesdropper would require to break into P2P application server, resulting in an extra security layer before access to the content itself. Also, while using a publicly and anonymously shared content,

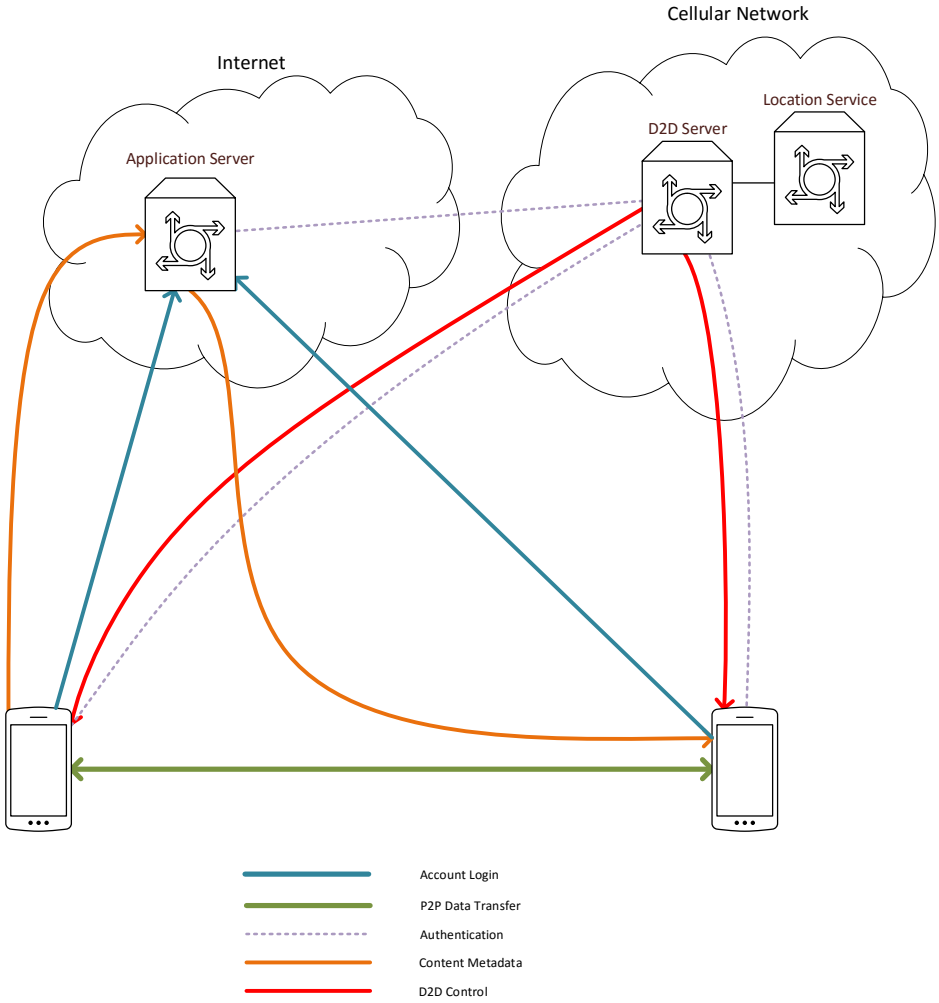


Figure 3.1 *Assisted D2D connection establishment via D2D server*

a peer does not obtain information about which user exactly is serving it, or what other services are shared by the same UE.

- There is no need for UEs to take the burden of peer discovery, by sending and listening to broadcast requests. UEs can rely on the cellular network to instruct them, when it is suitable to start D2D radio interface and with which parameters. This is utterly important, as keeping D2D interface active can consume significant amount of energy even in case when device does not need to send or receive data.

- The network can provide temporary link layer addresses to users, thus alleviating the need to reveal their actual IDs to each other. Devices do not need

to publish their actual IDs for discovery or connection setup.

- Network operator can monitor traffic rates and quality of service on network-assisted D2D connections. Even though it is not possible to monitor link states directly, the network can establish a control loop with UE to report and adjust transmission rates and QoS. Having this information, the network can plan and coordinate with existing infrastructure to properly allocate radio resources.

Suggested architecture of network-assisted D2D is easily integrated into existing 3GPP LTE deployment [31]. Figure 3.2 shows the architecture of the integrated system. Only one new entity has to be added to operator's core – the D2D server. Such placement also allows D2D server to receive information on UE location from eSMLC and to effectively interact with other entities on the Internet.

3.3 Implementing traffic offloading prototype

Within proof of concept implementation, the author of this work was responsible for making required changes to UE operating system, elaborating offloading algorithm based on IP routing, setting up and maintaining infrastructure to run prototype components, and implementing UE side of the prototype. Next subsections elaborate on implementation of these prototype components.

3.3.1 Android networking subsystem

Android, as a Linux-based system[1], allows to have simultaneous connections over more than one radio interface. Once LTE and WiFi connections are active, mobile device has two directly connected networks and one default gateway to send traffic outside of these networks. In this state, it is possible to reach other peer on WFD link only when destination address in IP packet header is WFD address of the peer, and source address is the one of originating mobile device. Also, WFD link uses private address range that is not reachable through anything else than WFD link; once the link is disconnected, the peer becomes unreachable. For the suggested D2D architecture, a device would be required to be able to reach peer's public IP address on LTE interface through WFD link.

[1]Android OS: http://www.openhandsetalliance.com/android_overview.html

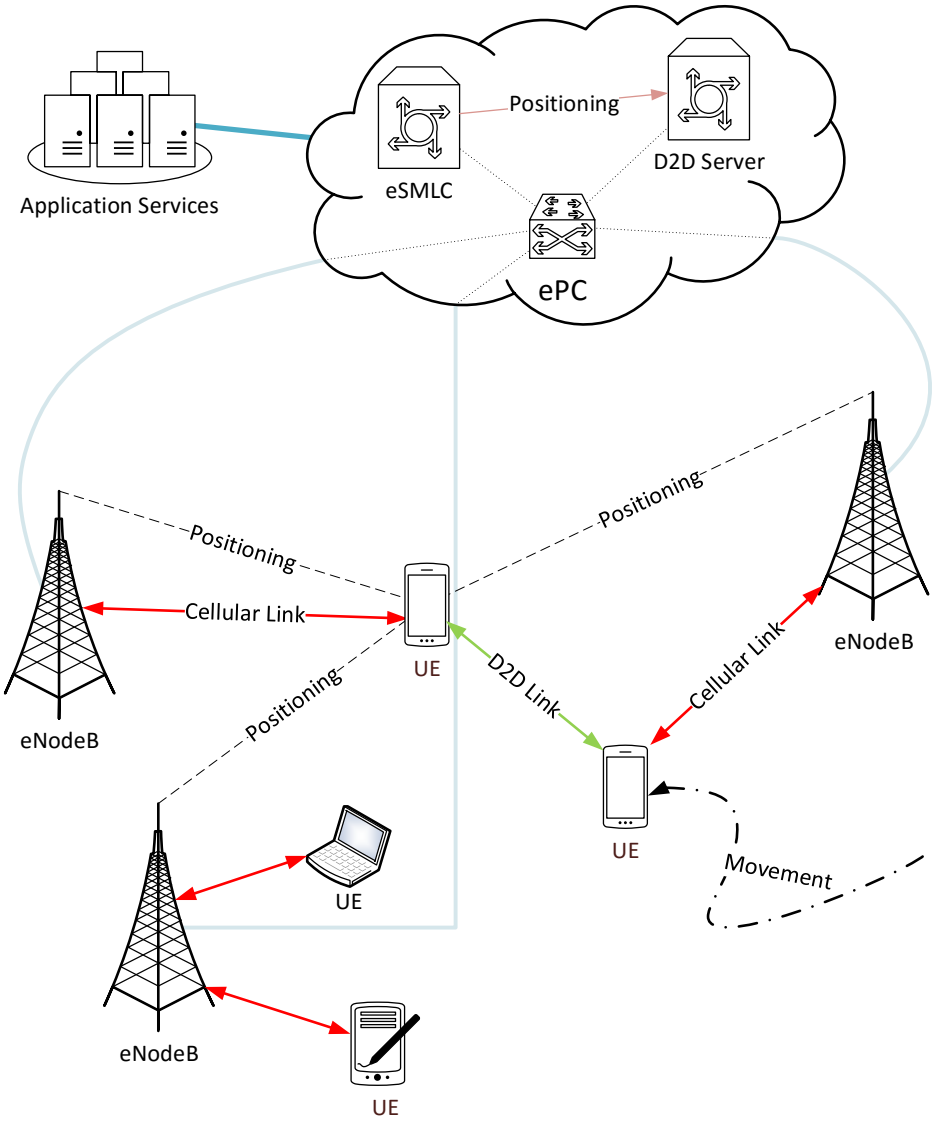


Figure 3.2 *Proposed D2D services layout*

One of the goals was to create a solution that would be transparent for already existing applications, and this way ease adoption of the technology. For this reason, modifications at the physical layer are not desirable, as it is tightly coupled with radio interface driver. The same applies to Link layer – implementing new functionality there would make the solution vendor-specific. Since existing applications heavily rely on existing transport layer protocols, modifications in transport layer are also not considered. On the other hand, system offers various tools to make changes in IP layer (or Network layer). This would allow to leave underlying radio interfaces as-is, and would enable application developers to use same routines to obtain network access as before. IP addresses are in a way bound to the physical interfaces, but selection process is made without direct interaction with interfaces. This would allow us to create an interface independent solution without modifications of upper layers.

The default configuration of an Android system allows to have routes with multiple gateways, but one of them is inserted into routing table with lower cost than the others. This way no load-balancing is performed and only one route is used. In case of WiFi link and LTE link, or cellular link in general, the route through LTE gateway is preferred for the Internet connectivity because WiFi link and, especially, WFD do not guarantee Internet connectivity at all. Therefore, changing cost of default route would cause unreachability of the Internet for all applications in the mobile device.

3.3.2 Traffic offloading based on routing

The proposed solution is based on allowing mobile device to route IP traffic as usual, and then inject more specific routes for particular peer into routing table. As Android system is based on Linux kernel, it is possible to enable routing functions by modifying the value of system variable net.ipv4.conf.all.forwarding from 0 to 1. After this change, Android mobile device gains capabilities of a generic router, known from computer networks. This way, mobile device can forward packets from one interface to another. The interesting point to note here is that it allows to send IP packets with source address being LTE interface public IP address and destination address being peer's WFD private IP address, and IP layer of Android system will send them through WiFi interface.

There are, however, several issues with such approach to link selection that need

to be solved. The first issue arises with application use of network sockets. When an application needs network connectivity, it requests the service from operating system using sockets API, and operating system chooses to use one of the available IP addresses as source address. Source IP address, source transport layer port, destination IP and destination port must remain the same throughout communication, which cannot be guaranteed in case session is bound to WiFi interface IP addresses, because interface can be disabled at any moment. Another issue is with routing private networks through operator's infrastructure – if a session is bound to WFD's private IPs it cannot be switched to LTE interface because the operator's infrastructure prohibits routing packets to this private IP range.

As a workaround for prototype demonstration, a set of OpenVPN tunnels has been elaborated from devices to an anchor point in the network infrastructure, thus allowing routing of private networks through operator's core. In a real world deployment, however, this would not be necessary, as operator could set up internal routing according to D2D link networks, issued by D2D server. Also, the system creates an *overlay* GRE tunnel bound to loopback interfaces on the devices. Loopback interface is meant to be a constant anchor point for applications, and only *overlay* tunnel endpoints will be rerouted, ensuring that session IP addresses remain the same all the time, regardless of *underlay* interfaces used, and thus providing service continuity.

Since only communication with one particular peer needs to be offloaded, a route can be inserted into routing table stating that just the peer's loopback IP address is reachable through peer's WFD private IP address. Insertion is performed by the command ip route add PEER_LOOPBACK_IP/32 via PEER_WFD_IP. Once this command is accepted by IP routing layer of Android, all traffic with destination IP address PEER_LOOPBACK_IP will be forwarded to WFD interface and this way *overlay* tunnel traffic will be sent over WFD.

The insertion and removal is performed by management application that is running in the background. Both actions are invoked upon a corresponding command from D2D server, but route removal can additionally rely on local channel quality measurements. Within this implementation, the only channel quality metric considered is the RSSI. When reaching a particular threshold in RSSI, route can be inserted or removed, thus selecting D2D or infrastructure channels, and reaching even higher threshold can trigger complete link disband.

This injection of specific routes into routing table does not have to be performed symmetrically on both devices participating in D2D offloading. Also, route injection scheme is not limited to be used only with a single peer.

3.3.3 Infrastructure for the prototype

The service relies on three parts: a) Client side phone application b) Content register service c) D2D server.

The content register is assumed to be any platform providing data sharing services to its users e.g., social networks like Facebook, Google+ etc., with a difference that in this case only metadata about content is shared in form of a hyper-reference, instead of actual data. D2D server, on its turn, is designed to be run by network service provider. It is worth noting that besides suggested architecture layout, the service is flexible enough for both entities to be run by a single authority or even on the same host.

To emulate integration of the service with providers, both server side applications were deployed within research group's cloud infrastructure. Two virtual machines were setup to act as Content Register and D2D server. Content register part was implemented in PHP, served by Apache web server. Hence, the application is a generic website that offers a registered user possibility to post their intent to share some data, or to search for shared data locations. User posts only information required to access data, rather than its location i.e., sharing protocol and port number, while IP address that locates the data, is handled by D2D server. The part that glues service sharing and service discovery is the introduced protocol identifier in URI returned by content register – "d2d://". End user devices were configured to interpret this protocol scheme as a request to start Client application that in turn is capable of communicating with D2D server to translate username of the peer serving data into an IP address.

D2D server was implemented in Python as a standalone application, using HTTPS as transport for control messages. System assumes user's mobile data link to be up all the time during service usage. Implemented solution shows that content register and D2D services are easily integrated in a seamless fashion into existing web serving infrastructure.

End user devices used in service demonstration were Sony Xperia ZL phones pro-

vided by Sony Mobile. Android, being an open source mobile platform, provides needed flexibility in configuration and available tools to fulfill requirements demanded from user devices. One of the main features user device should have is the ability to have both mobile data link and WiFi link to be up simultaneously. Due to energy consumption constraints, most of consumer devices on the market restrict their network connectivity to use only one of the available connections at a time. To enable both interfaces in the system, author had to bypass native Android service controlling WiFi, and interact with WiFi driver directly. As Android system to some extent is derived from GNU\Linux, it easily gives us needed tools – *wpa_supplicant* interface controlled via *wpa_cli* utility. Another issue is that in order to integrate with these utilities Android has to be compiled with *userdebug* feature enabled, and stock firmware provided by Sony for their devices does not have this property. Therefore, another option to consider was to build the system based on source code from Android Open Source Project, using proprietary binary drivers released to public by the vendor. However, resulting system at that time lacked radio drivers, needed for intended operation of mobile data connection. Final choice for the user device platform was popular aftermarket firmware Cyanogenmod[2], based on Android and maintained by FreeXperia group.

Another requirement for end user's mobile system is to be capable of receiving incoming connections through mobile data link. Considering the fact that most operators use private IPv4 address pools to assign to user devices and provide Internet connectivity by means of NAT, accessing the services running on users device from outside of its local link is not feasible with such setup. One of the possible solutions to overcome this issue would be using IPv6 addresses, but local service providers at the time being did not provide IPv6 connectivity options to the extent suitable for practical demonstration of D2D service. Another tested option for demonstration was encapsulating mobile data link of both communicating devices inside a VPN tunnel to a common VPN server, thus moving both devices into the same IP subnet. Also, discussing the issue with local service provider – TeliaSonera Finland Oyj.– researches were offered a certain access point name that would provide user devices with a publicly routable IPv4 address.

[2]Cyanogenmod distribution: https://wiki.cyanogenmod.org/w/Main_Page

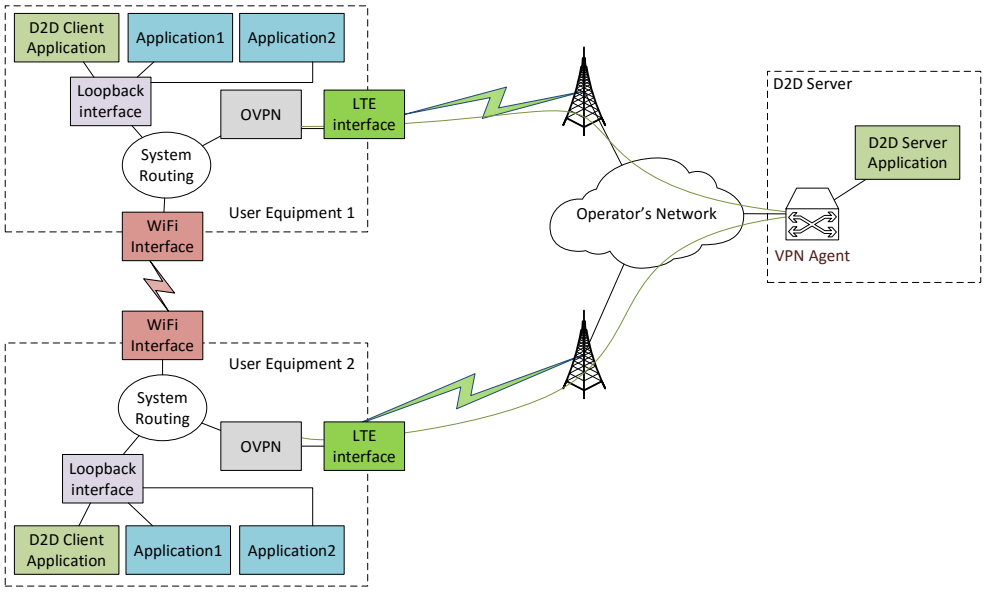


Figure 3.3 *Prototype setup for D2D architecture*

3.4 Prototype implementation technical details

This section covers deeper technical details of implementation for service architecture prototype described in sections above. As mentioned before, the whole system comprises three entities – UE, D2D server and Content register. Besides that, prototype infrastructure contains a node serving as anchor point for VPN layer, and a node serving as gateway for Internet connectivity.

3.4.1 User equipment

User side of the prototype was implemented in two layers – main network assisted offloading logic, implemented as native Android application, and functional layer performing all interactions with Android OS implemented in Bash shell scripts.

Bash scripts implement functions like inserting/withdrawing routes, enabling WFD links and establishing tunnels to the infrastructure. Also, they provide an abstraction layer to D2D Client application, while client application in turn establishes control channel with D2D server and interacts with the server using HTTPS to execute offloading algorithm steps.

The workflow of suggested solution starts with running D2D client Android ap-

plication. The first phase for application is to invoke initialization *shell* script d2d_init.sh (Program 4) to bring UE to a state, where it would be ready to interact with infrastructure and D2D server. As mentioned in subsection 3.3.3, system WiFi service will not allow to have both radio interfaces up simultaneously, so the initialization script will stop this service and will run a new copy of *wpa_ supplicant*, with configuration files and parameters needed to prepare WiFi *p2p* interface. Also, initialization script loads necessary additional kernel modules. The last phase of initialization script is to start OpenVPN tunnel, and this phase creates a blocking dependency, where initialization script in order to complete needs OpenVPN tunnel to successfully establish connection with infrastructure. This is resolved by implementing a *trap* in the initialization script, so that it can safely *sleep* after starting OpenVPN tunnel, and be resumed by *openvpn* process after tunnel is up. Finally, inititialization script will mark its completion in a process *pid* file, so that other system parts can later check if initialization has already been performed.

After *init* phase is done, mobile device has established a control channel to D2D server via the tunnel, and is ready to setup WiFi direct connections, when instructed to do so. D2D Client application regains control, registers itself to D2D Server, and UE becomes available for offloading. Depending on the role of UE as publisher or consumer, D2D Client application is invoked either manually or by a link from Content Register (see subsection 3.4.3) respectively. Since components responsible for the network-assisted offloading features are completely decoupled from application layer, in case user has published some content, they must ensure that the application that is actually serving it (e.g., a file server, game server or multimedia streaming application) is running and is awaiting for incoming connections. In case user followed the link from Content Register, D2D Client application subscribes to server with data from the link. At this stage, network knows that one user is ready to serve content and the other is willing to receive it with offloading features enabled.

The next step to prepare devices for traffic offloading is to start *loopback* interfaces and *overlay* GRE tunnels. When both peers public LTE (or OpenVPN in case of this implementation) IP addresses are known, D2D server instructs both devices to start a *loopback* interface, to add routes to peer's *loopback* interface over cellular network and to bring up a GRE tunnel bound to those *loopback* interfaces. These functions to operate *loopbacks* and GRE are implemented in corresponding *shell* scripts, see Program 5. *Overlay* tunnel endpoints will be rerouted to switch traffic from one radio network to another providing service continuity as described in sub-

section 3.3.2. Addressing for the *loopback* networks and *overlay* network is done by coupling predefined prefixes with host portion of devices LTE IP address.

When D2D server detects that two users, who have previously subscribed to be assisted in D2D communication, are now in appropriate conditions to use offloading, server will generate a group name and a pre-shared key for WFD, and will instruct one device to start WFD invoke a Bash *shell* script implementing start, join, leave, or remove functions. The script, in its turn, will use *wpa_cli* tool to interact with *wpa_supplicant* process started at initialization phase. As in case with *loopback* and *overlay* network addressing, for this prototype implementation, when WFD link is established it is configured in a stateless manner with IP addresses from 10.1.1.0/30 range with .1 assigned to group owner and .2 to WFD client.

At this point, both devices have two radio interfaces up and are ready to offload traffic, by rerouting *overlay* tunnel endpoints through either of these interfaces. When instructed by the network, D2D Client application will insert or withdraw routes by invoking corresponding *shell* scripts. D2D server keeps track of UE location, and when the distance and RSSI metrics reach predefined levels, server will instruct devices to use WFD, to fallback to using cellular network while keeping WiFi radio on, or to completely shutdown WiFi interface.

Upon users' request D2D server can de-register them and stop following their proximity.

3.4.2 D2D server

Server is running Python application listening for HTTPS requests on port 8099. All the relevant data from clients are passed as HTTP parameters and server is running an internal database to store registered users and their associated data like public IP addresses and current location. In present prototype implementation UE will periodically poll the server about status changes and server will reply with corresponding instructions according to current network state.

3.4.3 Content register

Content Register is a PHP application, running embedded database to authenticate users and store metadata about their shared content. When users wish to post some

service available for access and ready to be offloaded to D2D link when appropriate they log in to Content register and publish access protocol and port number for the service e.g., http/80. When another user sees this intent to share, they can request shared content, and Content Register will present the metadata in form of a URI with d2d:// scheme and embedded information about peer username, target D2D server, shared service protocol and port
d2d://amie@d2d.winter.rd.tut.fi@simhost.winter.rd.tut. fi /webcam:8080.

Web browser in UE is setup to recognize d2d:// URI scheme as bound to D2D Client local Android application. Hence, following the link given by Content Register will start D2D Client application that will establish control channel to its D2D server, and will register its intent to engage into D2D connection with the named peer, when network conditions are favorable to do so. Optionally, user may choose to start using shared content straight away and be offloaded when possible, or to wait before content is available via direct link.

3.4.4 Utility nodes

All server side components are running in virtual environment, implemented with vSphere from VMWare. Both D2D server and Content Register are running on dedicated virtual machines. To terminate OpenVPN connections, another dedicated VM was installed. The VPN VM is selectively routing traffic from VPN connections, either to other peers using *openvpn* "subnet" type of configuration, or to the Internet gateway using Policy Based Routing. The Internet gateway is another VM, running Brocade Vyatta[3] software router, and is performing NAT to provide Internet connectivity to internal networks.

3.5 Conclusions on network-assisted D2D communications

D2D connectivity emerges as an important enabler for traffic offloading from network infrastructure and allows network operators to implement proximity aware social networking. Elaborating a network-assisted D2D technology prototype has emphasized potential gains of using direct connectivity between closely located devices as well as highlighted out major challenges. Significant capacity gains at system level

[3]Vyatta Router: http://vyos.net/wiki/Main_Page

have been confirmed by previous research, most important findings after prototype development are summarized below.

- Using existing web application paradigm enables easy adoption of D2D connectivity features by application service provides. Some of the operators could benefit significantly by employing this new architecture to offer new services that would otherwise be impossible without continuous use of GPS positioning on the UEs.
- The example of integration of network-assisted D2D service into Android platform shows that same management protocols could be implemented by equipment manufacturers themselves. Certain platforms do not offer proper programming tools to give control of routing subsystem to developers, but this should not be an issue if vendor decides to implement the technology.
- D2D links offer much more capacity and lower latency than current cellular connections can provide. At shorter distances under 50 meters WFD can easily stream HD video or run real-time applications. Also, energy efficiency is very attractive.
- Currently, mobile operators' networks are heavily dependent on Network Address Translation (NAT) and sophisticated filtering. That complicates prototyping of new protocols in general.
- A vital requirement for entire proximity based offloading concept is support for positioning in operator's network. Appropriate standards are already being developed by 3GPP.

Prototype implementation improved research vision on D2D communications under certain assistance from infrastructure network. It confirmed that suggested architecture significantly improves user connectivity and enables a variety of new services. Building *overlay* network on top of physical links allowed for flexible traffic rerouting in a way seamless to the end users. Once again, virtualization proved to be a very useful tool in developing new architecture prototypes, allowing to deploy and ideas and research scenarios in a quick and agile way.

Figure 3.4 presents a short video introducing the technology[4] and available at http://winter-group.net/video-lte-assisted-wifi-direct/.

[4]Demo at MWC: http://winter-group.net/mwc_2014_d2d_demo/

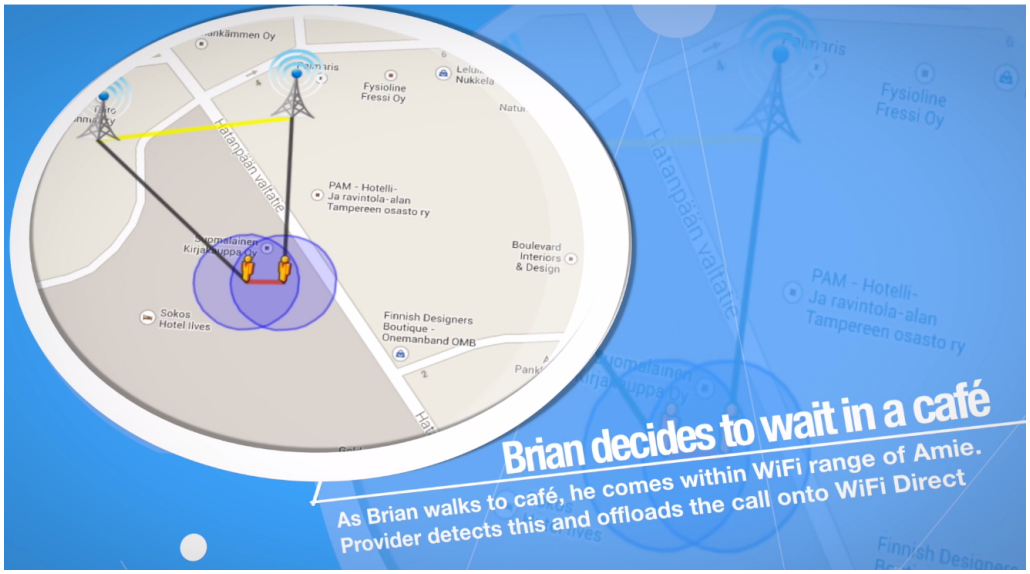


Figure 3.4 *D2D video presentation*

Achieved results suggest that within next few years challenges outlined during prototype implementation will be resolved and the first services offering these features should appear on the market shortly after. Demands for higher capacity and new services force network operators to seek new architectures and technologies, presented technology prototype is an easy to implement solution and is a decisive step forward. Also, with more D2D servers deployed additional ways of intelligent and centralized management of D2D connections will arise. That would potentially bring further improvement to capacity and efficiency. To prove feasibility of suggested technology, a full-scale practical trial of network-assisted WiFi-Direct has been completed on a live 3GPP LTE deployment in Brno, Czech Republic [31].

4. SECURITY AND TRUST IN PROPOSED ECOSYSTEM

To make the technologies described in previous chapters available for wider adoption, additional efforts have to be made to ensure security and privacy for the users. Chapter 3 introduced network-assisted D2D communication framework, a wireless technology allowing cellular infrastructure to control direct connectivity between proximate user devices. As this approach blends together centralized and distributed network architectures, it requires new solutions to enable secure, private, and trusted data exchange, even in cases when cellular control link is not available at all times. Additionally, the Internet of Things (IoT) communication framework aims to facilitate ecosystems of ubiquitous smart devices. Therefore, augmented with heterogeneous connectivity it would enable new kinds of services for a wide variety of domains like public transportation, education, health care, and public safety. Within this new paradigm of ubiquitous connectivity, security related issues become even more crucial.

4.1 Securing network-assisted D2D communications

The research covered in Chapter 3 outlined network-assisted WiFi-Direct D2D offloading technology, where unlicensed bands are used to relieve cellular network congestion basing on the fact that multiple radio interfaces (e.g., 3GPP LTE and WiFi) are available in today's mobile devices. The prototype developed for this work has been successfully integrated in live cellular core to demonstrate the feasibility of a PoC implementation of such technology [32]. With the approach of employing centralized LTE infrastructure to assist in formation of WFD D2D links, centralized assistance (control) is imposed over otherwise distributed communicating proximate pairs, thus combining centralized network architecture and distributed system architecture that were previously independent.

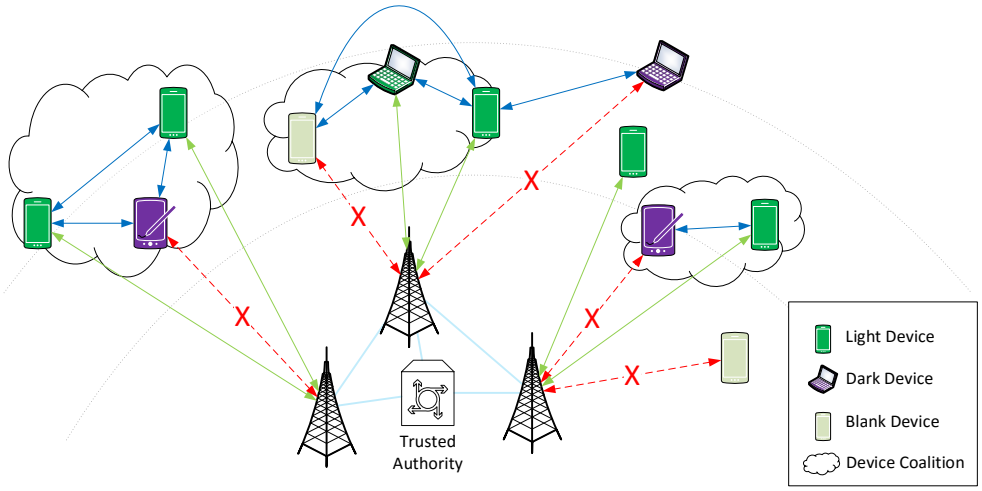


Figure 4.1 *Connectivity example for security framework*

"Light" device has a secure channel to central authority; "Dark" device has lost its connectivity to central authority; "Blank" device had never connected to authority but wishes to join the coalition.

Main target for research outlined in this section was to study the resulting hybrid centralized-distributed architectures, illustrated in Figure 4.1, in terms of secure data delivery for users already engaged in D2D communication, in conditions when cellular connection may become temporarily unavailable due to a variety of different factors, such as user mobility, obstacles, etc. [33]. When all peers are connected to centralized infrastructure, imposing their own information security (IS) rules is straightforward with conventional methods. However, in cases of intermittent LTE connectivity, proposed solution delegates a certain number of user devices in this group with ability to admit a new (previously unassociated) device, or to exclude one of the existing members from the group.

Currently, this way of admission and exclusion of devices in/from the group can only be managed by Public Key Infrastructure (PKI) deployed in the cellular network, while proposed algorithm extends this functionality for the cases of intermittent cellular connectivity e.g., in tunnels, airplanes, lifts, etc. After conducting a thorough state-of-the-art overview of potential security, privacy, and trust solutions, suitable for proximity-based communication, this research proposed novel information security protocol for network-assisted D2D connectivity and demonstrated its practical implementation. Figure 4.2 presents demonstration of the protocol operation. This protocol is able to operate even when cellular network connection becomes tem-

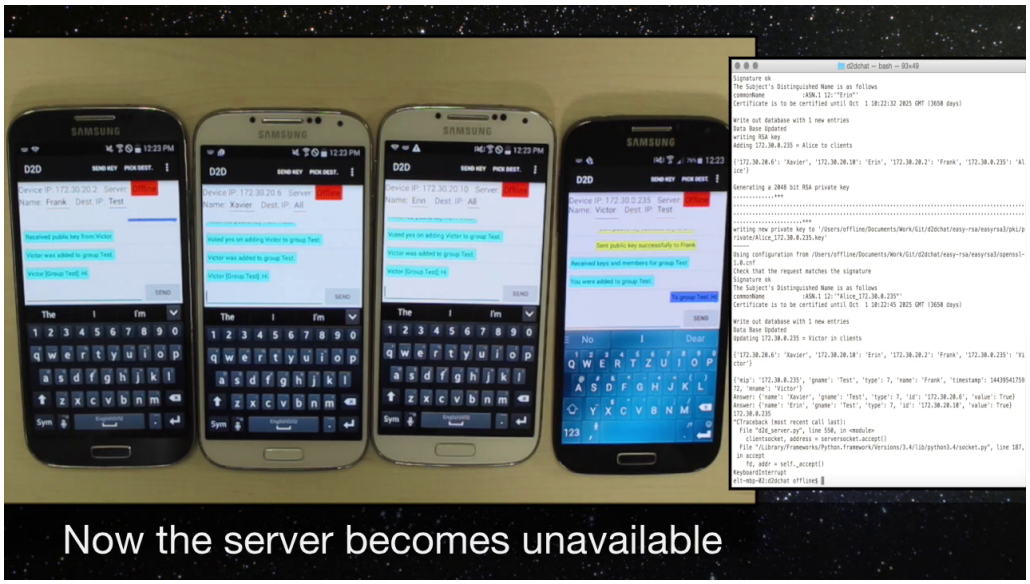


Figure 4.2 *Demonstration of security framework application*

Full video presenting the experiment is available at http://winter-group.net/d2d-security-lte-advanced-network/

porarily unavailable [7].

To lay foundation for a prototype implementation, the author was tasked to elaborate a system demonstrating major benefits of this novel security framework, and suggest example operations within it. The system was designed to include the following features:

1. The setup should define networks in such a way that user devices would be able to interact via both trusted and untrusted channels, where trusted network is the one with direct access to PKI authority.

2. Every UE must be uniquely addressable in a way that would allow other participants to communicate with it directly, without utilizing any intermediate device. It implies that data exchange should happen through direct connection between devices without relaying via a common exchange server.

3. UEs should be able to form coalitions sharing communication setup parameters. Such a coalition would define a logical group of securely-communicating devices.

4. The system should provide UE with functions to selectively admit or exclude users from the coalition. The decision should be made based on a consensus

among predefined subset of devices.

5. All system functions should stay valid even when PKI authority network becomes unavailable. End to end communications should maintain same security levels, as well as admission and exclusion of new devices should also comply with security policies previously defined by authority.

To fulfill the requirement in Feature 1 network setup for the new prototype inherited connectivity features from network-assisted D2D traffic offloading architecture described in Chapter 3. Characteristic features of D2D prototype align well with the suggested security framework. Requirement in Feature 2 was imposed to emphasize applicability of suggested security solution to any data streams, and make it easily portable to other existing communication protocols. Features 3, 4 and 5 are direct application of the suggested security framework.

The resulting solution was an instant messenger application developed for Android operating system. Abstracting from peer discovery functions, important features of this type of applications are ability to directly send data to a known peer, and to form groups with respective users. The users in the system were identified by their IP addresses, and messaging functionality was implemented directly over TCP sockets. This would allow to prove the concept generally for arbitrary TCP streams.

The work was taken forward by other research group members, to implement security functions and lead to a full scale prototype[1].

4.2 Characterization of cryptographic primitives in IoT

Many advanced cryptographic applications like identity-based encryption and privacy protection, rely on pairing-based schemes. The next step in research of security in heterogeneous environments focused on assessing smart phones and modern wearable devices like smart watches and embedded devices, and elaborating a comprehensive view on their viability to be used with state-of-the art cryptography utilizing bilinear pairing for real-time communication. To provide a broader view, results obtained for bilinear pairing algorithms were compared with performance evaluation results for hash functions, block cipher and digital signatures. The goal

[1] http://winter-group.net/d2d-security-lte-advanced-network/

of this research was to test if some devices are suited better for operations with certain cryptographic routines, and results show that wearable devices of today have enough computation potential to efficiently operate with cryptographic primitives in real time.

Highly heterogeneous ecosystems enabled by IoT interconnection framework comprise various communication modes like Human-to-Human (H2H), Human-to-Machine (H2M), and Machine-to-Machine (M2M) communications. Wearable technology, as part of IoT, raises the scale of technology diversification to unprecedented size. Within IoT vision, wearable devices are Internet connected "smart devices" built on microchips (System on the Chip, SoC), equipped with multiple sensors and wireless communications interfaces that operate in the immediate vicinity of their user [34]. These devices collect data for the user, track activities and improve user experience across different application domains.

Research on the wearable devices market performed by telecommunication industry leaders, such as Juniper [35] and Cisco [1], predict that global retail revenue from smart wearable devices will tend to reach $53.2 billion by 2019, compared to the $4.5 billion at the end of 2015. Smart watches and smart glasses are envisioned to be the dominant part of the wearables market over the following five years. Today, there are also many social and legal challenges to be solved for wearables, as it always happens with new and highly innovative digital technologies. Without the ability to run strong security frameworks on the devices, attacks and misuse of wearables might invalidate any of the expected benefits.

Therefore, a lot of research and engineering efforts are devoted to the topic of security features in the IoT era of ubiquitous wearable devices. To protect data transmitted over the network, wearable devices can use public key cryptography tools, like digital signature schemes providing user authentication and keys to encrypt the data. Main challenge for IS specialists is to design digital signature schemes that would comprise three features (i) security, (ii) computational efficiency, and (iii) small communication overheads. Mathematical formulations, such as the discrete logarithm problem, the RSA problem, or integer factorization [36], form the basis for standard operations in conventional digital signature schemes, thus providing standard security properties, including authenticity, integrity, and non-repudiation.

The research conducted at TUT addressed not only classical cryptography but also pairing-based algorithms, as well as evaluated their efficiency for wearables and other

Figure 4.3 *Devices used in assessments*

constrained IoT and user equipment like smart watches, smart phones, and embedded devices. Pairing-based cryptography is used to implement privacy-enhancing features that otherwise would be difficult to implement using conventional asymmetric cryptography in modern solutions. Bilinear pairing operations enable design schemes like group signatures [37, 38], anonymous attribute-based credentials [39, 40], or identity-based encryption [41]. Mechanisms of efficient revocation of invalid devices based on dynamic accumulators [42] and identification of attackers are particularly important for the IoT system operation, and would be difficult to construct without pairing-based cryptography. With this in mind, this research addressed and evaluated some of the common personal and wearable devices starting from the conventional smartphones (Samsung Galaxy S4, Apple iPhone 6, etc.) to the embedded devices (Intel Edison, Raspberry Pi 1 Model B, Raspberry Pi 2 Model B) to smart watches (Sony Smart Watch 3, Apple Watch), see Figure 4.3 for the summary of selected devices, in their ability to perform both standard and advanced cryptographic operations, as well as their efficiency with pairing-based cryptography primitives functionality. The author of this thesis was responsible for porting the test-suite to be run on embedded devices like Intel Edison.

Results of the tests demonstrated that operations of pairing-based cryptography can take from several hundreds of milliseconds to few seconds on current small-scale de-

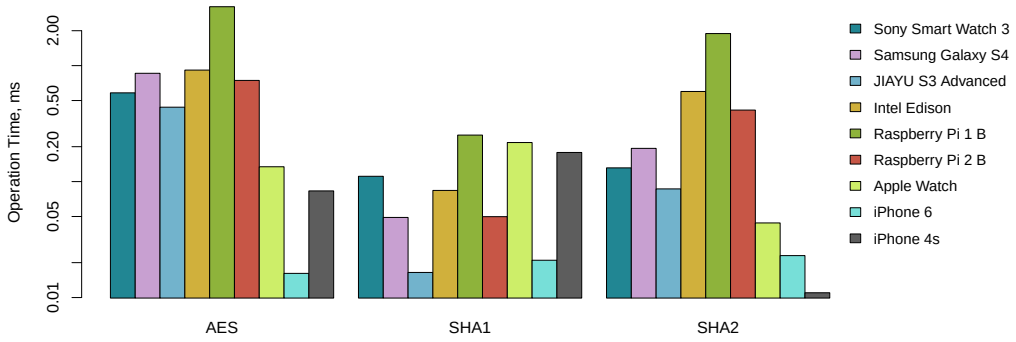


Figure 4.4 *Performance comparison for classic cryptographic primitives*

vices, with bilinear pairing operation being the most resource-consuming one. Other operations on the elliptic curves perform several orders of magnitude faster, and therefore a constrained device should execute only basic operations, while offloading pairing operations to a trusted device with more computation power [43, 44]. Also, applications that deal with real-time data transmissions should use classic cryptographic primitives (SHA2, AES, RSA) to enforce security, authenticity, and data integrity. Figure 4.4 presents assessment results obtained for classic cryptographic primitives.

The assessment demonstrated that computation power in modern wearable electronics has reached the level matching that of a two-year-old smartphone. Also a general purpose device like the Raspberry Pi showed lower performance compared to specialized IoT device like Intel Edison, even though the latter is rather designed for minimal energy consumption.

4.3 Applications of secure and trusted D2D connectivity

One of the most promising classes of network-assisted D2D applications is based on the situations, where a large number of people are brought together casually and may benefit from their proximity for business or pleasure, but at the same time require certain levels of security, privacy, and trust for their communication.

An example of such a scenario may involve a person (group of people) commuting on a public/intercity transport line and willing to interact with other passengers

by sharing some content, playing games, or getting involved into any other social activities, such as chat. A user may thus sign-in for its customizable commuter entertainment services well in advance and has a choice to also subscribe to a group package with friends and/or family members.

At a later time, whenever the client is in geographical proximity to the carrier, the system activates automatically to offer a personalized selection of user services throughout the journey. Importantly, together with securing the desired multimedia content, the mechanisms described in this work offer rich opportunities to facilitate proximal applications with other commuters, which may involve private and trusted unfamiliar connectivity. However, the unreliable connectivity situations may appear frequently during the trip, but the proposed solution is robust to sudden connectivity disruptions.

Solutions proposed in this chapter are meant to enable secure D2D communication in conditions of unfamiliar connectivity, limited access to the centralized authority, as well as can provide an entire palette of proximity-based policies on top of the sharing of data. For instance, on contemporary long-distance buses and planes, it has already become commodity to see small personal screens with a selection of videos, music, multiplayer games, news, and other contextual information. However, deployment of such systems incurs high market entry barriers, adds significant operational expenses, and is strongly tied to a centralized authority dispatching content and enforcing policies. Our solution allows people to use their own familiar handheld devices, whereas guaranteeing the desired levels of communication security.

5. CONCLUSIONS

This chapter concludes the thesis with a review of presented networking solutions for integrated heterogeneous wireless ecosystem. Wireless networks are constantly evolving in offered connectivity levels, thus strongly consolidating in our lives as a necessity. More and more devices are joining the network requesting continuous high quality service, which brings unprecedented challenges to network design in upcoming 5G era. Transformation of mobile user experience requires complex changes in both network infrastructure and device operation, where user experience is optimized taking into account surrounding network context. Primary focus of this work was to support current research on novel integrated multi-radio network architectures, by elaborating a set of prototypes and testbeds.

Solutions in Chapter 2 support research done on optimization problems in cooperative radio resource management in H-CRAN.

- Implementation is using COTS equipment making said resource management available already today, and demonstrating that no hardware changes are required.
- Prototype allows user device to be connected to both LTE and WiFi at the same time. Setting battery life issues aside, this feature significantly extends user connectivity.
- Resource allocation is controlled using multidimensional view on the network, employing non conventional parameters like cross RAT loading.
- System efficiently utilizes available radio connections by properly assigning traffic flows, e.g., streaming high resolution video over WiFi while downloading important software update over LTE.
- Deployment is flexible and loosely coupled to enable portability, automation and scalability of the solution.

Solution in Chapter 3 enables direct communication between proximate users to offload traffic from network infrastructure, and create new proximate services.

- Implemented using COTS equipment making the solution available without additional modifications to hardware.
- Allows to leverage peer proximity to offer richer set of user applications and services.
- Increases user awareness about proximate peers, services, and content. With this feature the user is not limited to only consume those services offered by the network provider.
- Network assistance ensures efficient use of short range radio interfaces by facilitating discovery and authentication functions. This significantly relaxes energy constraints currently seen in technologies like WiFi Direct.
- Deployment using conventional Web mechanisms enables prompt integration of new entities and connectivity solutions.

Applications in Chapter 4 support adoption of the above solutions by integrating a novel security framework, and assessing device readiness for efficient secure operations.

- Resulting security solution improves network assisted device-to-device communications and enables new type of secure proximity-based services and applications.
- Assessment of the common personal, wearable, and embedded devices provides insight into their readiness for secure operation in highly heterogenous wireless networks.

This thesis demonstrated application of novel and traditional networking concepts in implementing prototypes and demonstrators for emerging wireless network architectures and ecosystems. Resulted implementations supported research in centralized radio resource management, network assisted traffic offloading, and novel network security frameworks. Proposed approach demonstrated efficient prototyping for research scenarios using automation and abstraction concepts, decoupling network

functions from hardware, and modifying open source components. Use of modern software packaging and delivery techniques, like hardware level virtualization, OS kernel level virtualization, and automated configuration, significantly accelerated implementation process, and made resulting architecture components highly reusable for future projects.

Implemented prototypes and testbeds supported the corresponding research and resulted in several journal and conference publications, a book chapter and an industrial patent. Feasibility of implemented network assisted offloading architecture for device-to-device traffic was later validated during full-scale practical trial on a live network deployment.

BIBLIOGRAPHY

[1] Cisco, "Cisco Visual Networking Index : Global Mobile Data Traffic Forecast Update , 2015 - 2020," no. 4, 2016.

[2] M. Gerasimenko, D. Moltchanov, R. Florea, S. Andreev, Y. Koucheryavy, N. Himayat, S.-p. Yeh, and S. Talwar, "Cooperative Radio Resource Management in Heterogeneous Cloud Radio Access Networks," *IEEE Access*, vol. 3, pp. 397–406, 2015.

[3] M. Gerasimenko, D. Moltchanov, R. Florea, N. Himayat, S. Andreev, and Y. Koucheryavy, "Prioritized Centrally-Controlled Resource Allocation in Integrated Multi-RAT HetNets," in *2015 IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1–7, IEEE, may 2015.

[4] A. Pyattaev, K. Johnsson, A. Surak, R. Florea, S. Andreev, and Y. Koucheryavy, "Network-assisted D2D communications: Implementing a technology prototype for cellular traffic offloading," in *2014 IEEE Wireless Communications and Networking Conference (WCNC)*, vol. 4, pp. 3266–3271, IEEE, apr 2014.

[5] A. Pyattaev, O. Galinina, K. Johnsson, A. Surak, R. Florea, S. Andreev, and Y. Koucheryavy, *Smart Device to Smart Device Communication*. Springer International Publishing, 2014.

[6] A. Ometov, P. Masek, L. Malina, R. Florea, J. Hosek, S. Andreev, J. Hajny, J. Niutanen, and Y. Koucheryavy, "Feasibility Characterization of Cryptographic Primitives for Constrained Wearable IoT Devices," in *The First IEEE International Workshop on Security, Privacy and Trust for IoT*, 2016.

[7] A. Ometov, K. Zhidanov, S. Bezzateev, R. Florea, S. Andreev, and Y. Koucheryavy, "Securing Network-Assisted Direct Communication: The Case of Unreliable Cellular Connectivity," in *2015 IEEE Trustcom/BigDataSE/ISPA*, pp. 826–833, IEEE, aug 2015.

[8] J. Andrews, S. Buzzi, W. Choi, S. Hanly, A. Lozano, A. Soong, and J. Zhang, "What Will 5G Be?," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1065–1082, 2014.

[9] Y. Yang and T. Quek, "Optimal Subsidies for Shared Small Cell Networks - A Social Network Perspective," *IEEE Journal of Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 690–702, 2014.

[10] O. Galinina, S. Andreev, M. Gerasimenko, Y. Koucheryavy, N. Himayat, S. Yeh, and S. Talwar, "Capturing Spatial Randomness of Heterogeneous Cellular/WLAN Deployments with Dynamic Traffic," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1083–1099, 2014.

[11] B. Bangerter, S. Talwar, R. Arefi, and K. Stewart, "Networks and devices for the 5G era," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 90–96, 2014.

[12] X. Wang, *C-RAN: the road towards green RAN*, 2010.

[13] S.-H. Park, O. Simeone, O. Sahin, and S. Shamai, "Fronthaul Compression for Cloud Radio Access Networks: Signal processing advances inspired by network information theory," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 69–79, 2014.

[14] J. Li, M. Peng, A. Cheng, Y. Yu, and C. Wang, "Resource Allocation Optimization for Delay-Sensitive Traffic in Fronthaul Constrained Cloud Radio Access Networks," *IEEE Systems Journal*, 2014.

[15] Y. Zhou and W. Yu, "Optimized Backhaul Compression for Uplink Cloud Radio Access Network," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, pp. 1295–1307, 2014.

[16] Y. Yang, T. Quek, and L. Duan, "Backhaul-Constrained Small Cell Networks: Refunding and QoS Provisioning," *IEEE Transactions on Wireless Communications*, vol. 13, no. 9, pp. 5148–5161, 2014.

[17] S. Yan, W. Wang, Z. Zhao, and A. Ahmed, "Investigation of cell association techniques in uplink cloud radio access networks," *Transactions on Emerging Telecommunications Technologies*, 2014.

[18] M. Peng, S. Yan, and V. Poor, "Ergodic Capacity Analysis of Remote Radio Head Associations in Cloud Radio Access Networks," *IEEE Wireless Communications Letters*, vol. 3, no. 4, pp. 365–368, 2014.

[19] Z. Ding and V. Poor, "The Use of Spatially Random Base Stations in Cloud Radio Access Networks," *IEEE Signal Processing Letters*, vol. 20, no. 11, pp. 1138–1141, 2013.

[20] A. Liu and V. Lau, "Joint Power and Antenna Selection Optimization in Large Cloud Radio Access Networks," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1319–1328, 2014.

[21] M. Peng, Y. Li, J. Jiang, J. Li, and C. Wang, "Heterogeneous Cloud Radio Access Networks: A New Perspective for Enhancing Spectral and Energy Efficiencies," *IEEE Wireless Communications*, vol. 21, no. 6, pp. 126–135, 2014.

[22] *3GPP LTE Release 10 & beyond (LTE-Advanced).*

[23] X. Wu, S. Tavildar, S. Shakkottai, T. Richardson, J. Li, R. Laroia, and A. Jovicic, "FlashLinQ: a synchronous distributed scheduler for peer-to-peer ad hoc networks," in *Proc. of the 48th Annual Allerton Conference on Communication, Control, and Computing*, pp. 514–521, 2010.

[24] Qualcomm research, "LTE Direct The Case for Device-to-Device Proximate Discovery," tech. rep., Qualcomm, 2013.

[25] A. Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy, "Proximity-Based Data Offloading via Network Assisted Device-to-Device Communications," in *Proc. of the IEEE Vehicular Technology Conference (VTC-Spring)*, 2013.

[26] A. Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy, "3GPP LTE Traffic Offloading onto WiFi Direct," in *IEEE Wireless Communications and Networking Conference*, 2013.

[27] "Wi-Fi Peer-to-Peer (P2P) Technical Specification, ver. 1.1," tech. rep., Wi-Fi Alliance Technical Committee, P2P Task Group, 2010.

[28] 3GPP, "TR 23.890 Optimized Offloading to WLAN in 3GPP-RAT mobility," tech. rep., 3GPP, 2012.

[29] G. Fodor, E. Dahlman, G. Mildh, S. Parkvall, N. Reider, G. Miklós, and Z. Turányi, "Design aspects of network assisted device-to-device communications," *IEEE Communications Magazine*, vol. 50, pp. 170–177, Mar. 2012.

[30] J. Andrews, *Can cellular networks handle 1000x the data?*, 2011.

[31] A. Pyattaev, J. Hosek, K. Johnsson, R. Krkos, M. Gerasimenko, P. Masek, A. Ometov, S. Andreev, J. Sedy, V. Novotny, and Y. Koucheryavy, "3GPP LTE-assisted Wi-Fi-direct: Trial implementation of live D2D technology," *ETRI Journal*, vol. 37, pp. 877–887, 10 2015.

[32] S. Andreev, Y. Koucheryavy, J. Hosek, and K. Johnsson, "LTE-Assisted WiFi Direct Trial," *Global Communications Newsletter*, vol. 4, p. 3, April 2015.

[33] G. Fodor, S. Parkvall, S. Sorrentino, P. Wallentin, Q. Lu, and N. Brahmi, "Device-to-Device Communications for National Security and Public Safety," *IEEE Access*, pp. 1510–1520, 2014.

[34] A. D. Thierer, "The Internet of Things and Wearable Technology: Addressing Privacy and Security Concerns without Derealing Innovation," *Rich. JL & Tech.*, vol. 21, pp. 6–15, 2015.

[35] M. S. Whitcup and K. LaMattina, "Juniper – What is Inhibiting Growth in the Medical Device Wearable Market?." `http://bit.ly/1Dfffbf`, September 2014.

[36] L. Malina, J. Hajny, and V. Zeman, "Usability of pairing-based cryptography on smartphones," in *Proc. of 38th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 617–621, IEEE, 2015.

[37] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology–CRYPTO 2004*, pp. 41–55, Springer, 2004.

[38] L. Nguyen and R. Safavi-Naini, "Efficient and provably secure trapdoor-free group signature schemes from bilinear pairings," in *Advances in Cryptology-ASIACRYPT 2004*, pp. 372–386, Springer, 2004.

[39] C. Paquin and G. Zaverucha, "U-prove cryptographic specification v1. 1," tech. rep., Microsoft Technical Report, `http://connect.microsoft.com/site1188`, 2011.

[40] J. Hajny, P. Dzurenda, and L. Malina, "Attribute-based credentials with cryptographic collusion prevention," *Security and Communication Networks*, 2015.

[41] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Advances in Cryptology–CRYPTO 2001*, pp. 213–229, Springer, 2001.

[42] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," in *Advances in Cryptology—CRYPTO 2002*, pp. 61–76, Springer, 2002.

[43] R. Spreitzer and J.-M. Schmidt, "Group-signature schemes on constrained devices: the gap between theory and practice," in *Proceedings of the First Workshop on Cryptography and Security in Computing Systems*, pp. 31–36, ACM, 2014.

[44] J. Camenisch and A. Lysyanskaya, "Signature schemes and anonymous credentials from bilinear maps," in *Advances in Cryptology–CRYPTO 2004*, pp. 56–72, Springer, 2004.

APPENDIX A. CODE LISTINGS

```
port 1194
dev tun

#local 130.230.141.218

tls-server
ca /etc/openvpn/keys/ca.crt
cert /etc/openvpn/keys/wifi-vpn.crt
key /etc/openvpn/keys/wifi-vpn.key
dh /etc/openvpn/keys/dh1024.pem

mode server
topology subnet

route-gateway 10.219.0.1
ifconfig 10.219.0.1 255.255.255.0

ifconfig-pool 10.219.0.4 10.219.0.254 255.255.255.0

route 10.219.0.0 255.255.255.0
#push "route 10.219.0.0 255.255.255.0"
#push "route 10.0.0.220 255.255.255.255"
#push "redirect-gateway"

keepalive 10 60
inactive 1000

# The server doesn't need privileges
user openvpn
group openvpn

# Keep TUN devices and keys open across restarts.
persist-tun
persist-key

verb 4

route 172.16.1.0 255.255.255.0
#route 10.219.42.0 255.255.255.0
client-to-client

client-config-dir /etc/openvpn/3gpp-clients

up add_pbr.sh

client-connect /etc/openvpn/client_connect.sh
script-security 4
```

Program 1 *Open VPN configuration for LTE anchor*

```
#!/bin/env bash
systemctl stop wpa_supplicant.service

/usr/lib/connman/test/test-connman connect cellular_xxxxxxx_context1  #Sonera Roman

/usr/sbin/wpa_supplicant_p2p -B -Dnl80211 -f /var/log/wpa_supplicant_wfd.log -i wlan0 -c /etc
    /wpa_supplicant/wpa_supplicant.conf -O /var/run/wpa_supplicant -P /var/run/
    wpa_supplicant.pid &

echo "Waiting for wpa_supplicant..."
sleep 7

ip addr add 10.0.88.123/24 dev wlan0
ip link set up dev wlan0
ip route add 130.230.141.218/32 via 10.0.88.1 dev wlan0
ip route add 130.230.141.219/32 dev rmnet0

if ! lsmod | grep -q gre; then
```

```
    insmod /home/nemo/modules_3.4.91/gre.ko
    insmod /home/nemo/modules_3.4.91/ip_gre.ko
    insmod /home/nemo/modules_3.4.91/veth.ko
    insmod /home/nemo/modules_3.4.91/openvswitch.ko
fi

openvpn --remote 130.230.141.218 --ca /etc/openvpn/keys/ca.crt --cert /etc/openvpn/keys/jolla2.crt\
 --key /etc/openvpn/keys/jolla2.key --ns-cert-type server --nobind --persist-key --client --script-security 2\
 --up /usr/lib/connman/scripts/openvpn-script --up-restart --setenv CONNMAN_BUSNAME :1.71 --setenv CONNMAN_INTERFACE net.connman.Task\
 --setenv CONNMAN_PATH /task/1 --dev wifi-vpn --dev-type tun --persist-tun --route-noexec --ping-restart 0 --daemon

openvpn --remote 130.230.141.219 --ca /etc/openvpn/keys/ca.crt --cert /etc/openvpn/keys/jolla2.crt\
 --key /etc/openvpn/keys/jolla2.key --ns-cert-type server --nobind --persist-key --client --script-security 2\
 --up /usr/lib/connman/scripts/openvpn-script --up-restart --setenv CONNMAN_BUSNAME :1.71 --setenv CONNMAN_INTERFACE net.connman.Task\
 --setenv CONNMAN_PATH /task/0 --dev lte-vpn --dev-type tun --persist-tun --route-noexec --ping-restart 0 --daemon

echo "Waiting for openvpn..."
sleep 10
ip route add 10.119.0.0/30 dev lte-vpn
ip route add 10.118.0.0/30 dev wifi-vpn
echo 0 > /proc/sys/net/ipv4/conf/wifi-vpn/rp_filter
echo 0 > /proc/sys/net/ipv4/conf/lte-vpn/rp_filter

ip link add wifi-flow address 02:00:00:00:00:08 type gretap local 10.218.0.4 remote 10.118.0.2 dev wifi-vpn
ip link add lte-flow address 02:00:00:00:00:09 type gretap local 10.219.0.4 remote 10.119.0.2 dev lte-vpn
ip link set up wifi-flow
ip link set up lte-flow

ip link add veth0 type veth peer name veth1
ip link set address 02:00:00:00:00:01 dev veth1
ip link set address 02:00:00:00:00:00 dev veth0

ip link set mtu 1462 dev veth1
ip link set mtu 1462 dev veth0

ip link set up veth1
ip link set up veth0
ip addr add 192.168.219.2/30 dev veth0

ovsdb-server --remote=punix:/usr/local/var/run/openvswitch/db.sock \
 --remote=db:Open_vSwitch,Open_vSwitch,manager_options \
 --private-key=db:Open_vSwitch,SSL,private_key \
 --certificate=db:Open_vSwitch,SSL,certificate \
 --bootstrap-ca-cert=db:Open_vSwitch,SSL,ca_cert \
 --pidfile --detach

 ovs-vswitchd --pidfile --detach

ovs-vsctl del-port hetnet0 wifi-flow
ovs-vsctl del-port hetnet0 lte-flow
ovs-vsctl del-port hetnet0 veth1

ovs-vsctl add-port hetnet0 wifi-flow -- set Interface wifi-flow ofport_request=8
ovs-vsctl add-port hetnet0 lte-flow -- set Interface lte-flow ofport_request=9
ovs-vsctl add-port hetnet0 veth1 -- set Interface veth1 ofport_request=1

echo "Done."
```

Program 2 *Setup of Virtual Forwarding Layer with Open vSwitch*

```
#!/usr/bin/env bash

service openvswitch start
echo "Clearing containers..."
```

```
5 docker rm `docker ps -a -q`

7 echo "Starting VPN containers..."
  docker run --net=none  -P  --privileged=true --name wifi -d \
9     -v /opt/intel-het-net-d2d-demo/scripts/wifi-agent:/etc/openvpn openvpn /bin/sh -c "
          openvpn --config /etc/openvpn/default.conf"

11 docker run --net=none  -P  --privileged=true --name lte -d \
      -v /opt/intel-het-net-d2d-demo/scripts/3gpp-agent:/etc/openvpn openvpn /bin/sh -c "
          openvpn --config /etc/openvpn/default.conf"
13
  WIFI=`docker inspect -f '{{.State.Pid}}' wifi`
15 if [ -z $WIFI ]
     then
17     echo "ERROR: No WIFI container"
       exit 2
19 fi

21
  LTE=`docker inspect -f '{{.State.Pid}}' lte`
23 if [ -z $LTE ]
     then
25     echo "ERROR: No LTE container"
       exit 2
27 fi

29
  echo "Creating namespaces..."
31 mkdir -p /var/run/netns/
  ln -s /proc/$WIFI/ns/net /var/run/netns/wifins
33 ln -s /proc/$LTE/ns/net /var/run/netns/ltens

35
  echo "Preparing interfaces..."
37 ip link set wifi netns wifins
  ip netns exec wifins ifup wifi
39 #ip netns exec wifins ip route show
  #ip netns exec wifins ip route add default via 130.230.141.212
41
  ip link set lte netns ltens
43 ip netns exec ltens ifup lte
  #ip netns exec ltens ip route show
45 #ip netns exec ltens ip route add default via 130.230.141.212

47
  ip link add host-wifi type veth peer name wifi-host
49 ip link add host-lte type veth peer name lte-host

51 ip link set wifi-host netns wifins
  ip link set lte-host netns ltens
53
  ip addr add 10.118.0.2/30 dev host-wifi
55 ip netns exec wifins ip addr add 10.118.0.1/30 dev wifi-host
  ip addr add 10.119.0.2/30 dev host-lte
57 ip netns exec ltens ip addr add 10.119.0.1/30 dev lte-host

59 ip link set up host-wifi
  ip netns exec wifins ip link set up wifi-host
61 ip link set up host-lte
  ip netns exec ltens ip link set up lte-host
63
  ip route add 10.218.0.0/24 via 10.118.0.1 dev host-wifi
65 ip route add 10.219.0.0/24 via 10.119.0.1 dev host-lte

67

69 echo "Creating GRE tunnels..."
  ip link add wifi-flow address 02:00:00:00:01:08 type gretap local 10.118.0.2 remote
      10.218.0.4 dev host-wifi
71 ip link add lte-flow address 02:00:00:00:01:09 type gretap local 10.119.0.2 remote 10.219.0.4
       dev host-lte

73 ip link set up wifi-flow
  ip link set up lte-flow
75
```

```
ip link add hetnet type veth peer name host-flow
ip link set address 02:00:00:00:01:01 dev hetnet
ip link set address 02:00:00:00:01:00 dev host-flow

ip link set up hetnet
ip link set up host-flow

ip addr add 192.168.219.1/30 dev hetnet

ovs-vsctl del-port vswitch0 wifi-flow
ovs-vsctl del-port vswitch0 lte-flow
ovs-vsctl del-port vswitch0 host-flow

ovs-vsctl add-port vswitch0 wifi-flow -- set Interface wifi-flow ofport_request=8
ovs-vsctl add-port vswitch0 lte-flow -- set Interface lte-flow ofport_request=9
ovs-vsctl add-port vswitch0 host-flow -- set Interface host-flow ofport_request=1

echo "Done."
```

Program 3 *Setup of VPN processes as containers*

```
#!/system/xbin/env bash


# Script blocks waiting for signal
# that tun0 interace is ready
trap sigusr1 SIGUSR1

#Cleanup procedure makes sure stuck sleep gets killed correctly
sleep_pid=
trap cleanup EXIT

cleanup()
{
    [[ $pid ]] && kill $sleep_pid
    echo "Cleanup done"
}

sigusr1()
{
 echo -e "Waking up..."
 ip addr sh dev tun0
 echo 1 > /tmp/d2d_init_done
 exit
}


if [ -e /tmp/d2d_init_done ]
then
    echo "Init already done"
    exit 0
fi
svc wifi disable
if [ $? -ne 0 ]; then
    echo "wifi service disable fail"
    exit 1
fi
sleep 1

modprobe wlan
if [ $? -ne 0 ]; then
    echo "module insert fail"
    exit 1
fi

modprobe ip_gre

sysctl -w net.ipv4.conf.all.forwarding=1

cd /data/misc/wifi

/system/bin/wpa_supplicant  -iwlan0 -Dnl80211\
 -c/data/misc/wifi/d2d_supplicant.conf -N -ip2p0 -Dnl80211\
 -c/data/misc/wifi/p2p_supplicant.conf -puse_p2p_group_interface=1 &
```

```
55 if [ $? -ne 0 ]; then
    echo "wpa_supplicant start fail"
57  exit 1
   fi
59
   openvpn --daemon --config /etc/openvpn/3gpp.ovpn
61 echo -e "Waiting for tunnel interface"
   sleep 10000 &
63 sleep_pid="$!"
   wait
```

Program 4 *Initialization script on UE*

```
1 #!/system/xbin/env bash

3 ARGS=1

5 if [ "x$#" != "x$ARGS" ]; then
    echo "Usage: $0 remote 3gpp ip"
7   exit 3
   fi
9

11 remote_subnet=10.108.42
   gre_subnet=172.16.42
13
   remote_host=`echo $1| cut -d'.' -f4`
15 loopback_remote=$remote_subnet.$remote_host
   gre_remote=$gre_subnet.$remote_host
17
   ip route add $loopback_remote/32 via $1 metric 20
19
   loopback_local=`ip -f inet -o addr show d2d0 scope global |tr "/" " " | cut -d" " -f7`
21 local_host=`echo $loopback_local| cut -d'.' -f4`
   gre_host=$gre_subnet.$local_host
23
   ip tunnel add gre1 mode gre remote $loopback_remote local $loopback_local ttl 250
25 ip link set mtu 1476 dev gre1
   ip link set up dev gre1
27 ip addr add $gre_host/24 dev gre1
```

Program 5 *Starting overay tunnels*