Network-Assisted D2D Communications: Implementing a Technology Prototype for Cellular Traffic Offloading

Alexander Pyattaev†, Kerstin Johnsson⋆, Adam Surak†, Roman Florea†, Sergey Andreev†, and Yevgeni Koucheryavy†
†Tampere University of Technology, Tampere, Finland; ⋆Intel Corporation, Santa Clara, CA, USA
Email: {alexander.pyattaev, adam.surak, roman.florea, sergey.andreev}@tut.fi, kerstin.johnsson@intel.com, yk@cs.tut.fi

***Abstract*—Currently, cellular operators are struggling to relieve congestion on their networks in the face of rapidly growing mobile data traffic. While deploying an increasing number of base stations is expected to mitigate the disproportion between user demand and available radio resources, this solution is costly and plagued with many practical challenges. An attractive alternative is to enable cellular traffic offloading onto device-to-device (D2D) connections in the unlicensed bands, as current multi-radio user devices are already capable of establishing concurrent LTE and WiFi links. However, WiFi lacks a fast, efficient method of device/service discovery, and it is not equipped to efficiently manage numerous D2D connections. In our research, we have found that a limited amount of network assistance for D2D communications can overcome these limitations; and in this paper we describe our network-assisted D2D technology prototype. Specifically, we outline a complete standards-compliant solution that provides a seamless D2D connectivity experience to the end user. Our solution utilizes WiFi Direct as the link-layer technology for proximal D2D connections. However, the challenges faced during the design phase are universal to all D2D link-layer protocols, thus the proposed solutions are applicable to other potential D2D technologies.**

I. Introduction and motivation

Given that the proportion of mobile traffic transmitted over broadband wireless networks is expected to grow significantly in the very near future (i.e., 13-fold increase between 2012 and 2017) [1], it is commonly believed that even the state-of-the-art fourth generation (4G) technologies are very likely to face serious overloads [2]. Consequently, they will still be insufficient to meet the anticipated acceleration in traffic demand fueled by the rapid proliferation in types and numbers of wireless devices [3]. It is generally believed that the best way to address growing traffic loads is to reduce cells sizes and increase their density [4]. However, increasing the number of serving stations implies more complex interference management as well as higher rental fees and infrastructure maintenance costs [5]. Alternatively, a more flexible topology may be used, such as those proposed for client relay schemes, where users are also acting as relays for each other (such opportunities have been discussed before, for example in works [6], [7]). As a transition measure, however, to reduce the cellular operator's reliance on smaller denser cells, we propose that, whenever possible, neighboring client devices communicate via device-to-device (D2D) links instead of cellular links.

Over the last decade, a great deal of time has been spent defining a licensed-band D2D technology for Long Term Evolution (LTE) [8] by the Third Generation Partnership Project (3GPP) as well as by individual companies via proprietary solutions such as FlashLinq [9], [10]. Unfortunately, as always the standardization process takes time, so we do not expect to see devices equipped with licensed-band D2D on the market for several years. By contrast, unlicensed-band D2D technologies already exist on most client devices today. Unfortunately, while these technologies are time-proven and robust, certain key engineering issues remain. The primary limitation of unlicensed-band D2D technologies is the lack of efficient device discovery and connection management [11], [12]. Given limited battery lifetime on client devices, the power needed for device discovery and D2D connection establishment is unacceptable to most users. In addition, with today's "mobile" lifestyle, that lack of efficient, robust service continuity limits the kinds of services a user will engage in over D2D.

To address these issues, we propose a measure of network management for unlicensed-band D2D connections. For this purpose, we envision that client devices will be continually associated with their cellular networks and use this connectivity to help manage their D2D connections. In our recent work [13], [14], we demonstrated how network-assisted D2D can significantly increase cellular network capacity. In this paper, we show how the user's QoS perception may also benefit from this technology and present a potential development path for user equipment (UE) manufacturers and service providers. We expect the standardized solution for network assistance to be agnostic to the actual D2D technology used. Thus, since WiFi Direct [15] is the only D2D technology available on the market today that supports mid-range, high data rate P2P traffic (and in fact WiFi is already considered for cellular traffic offloading [16]), we concentrate on this attractive option in what follows. This work is primarily based on the 3GPP study item on D2D communications and proximate services (ProSe) [17].

The **primary contributions** of this paper are: (i) the 3GPP-compatible *architecture* for network-assisted D2D offloading, which is compliant with current web standards, and (ii) the complete D2D technology *prototype*, which was proven to work in real-world conditions providing reliable cellular (e.g., LTE) traffic offloading onto WiFi Direct links. The rest of this paper is structured as follows. First, we provide details of network-assisted D2D, explaining how the technology fits into the existing 3GPP LTE infrastructure and how it interacts with the third party applications. In Section III, we detail the actual

978-1-4799-3083-8/14/$31.00 ©2014IEEE

technical solutions implemented in our network-assisted D2D prototype and outline the major challenges we faced and how they were resolved. Finally, in Section IV, we discuss some interesting observations with regard to the QoS performance of the prototype.

II. The role of D2D assistance

Short-range communications provide higher data rates, lower transfer delays, and better power efficiency [12]. They also enable a significant increase in spatial reuse. Cellular network operators have taken advantage of these benefits by deploying increasing numbers of pico- and femto-cells in their service areas. However, they are missing a significant opportunity for capacity improvement when it comes to direct communication between clients. Currently, most client devices are equipped with direct communication capabilities, but the present technology is not suited to the relevant use cases for "proximity services". Interestingly enough, deployed technologies are not even capable of seamless offloading of existing cellular connections (e.g. mobile IP, although claiming such capability, is not actually utilized). Fortunately, with a limited amount of network assistance, these issues can be resolved in an efficient manner. Thus, network-assisted direct communication could be a logical "next step" to improving spatial reuse towards the vision of 1000x capacity [18] by the year 2020 in 5G systems.

D2D communication has received significant attention both in industry and academia due to the growing number of services and applications that can leverage proximity-oriented communications. The potential applications of D2D connectivity in cellular networks include local voice service (offloading calls between proximate users), multimedia content sharing, gaming, group multicast, context-aware applications, public safety, and more. Naturally, it is very difficult to guarantee a specific QoS on D2D links, since their channel qualities may vary significantly over time. As a result, delay-tolerant services such as file downloads have often been considered the best candidates for offloading. However, when clients are semi-stationary and in very close range, D2D communication is an option for many other P2P services, such as video streaming, social multiplayer games, etc.

D2D communication requires two basic operations: device discovery and connection establishment. While existing D2D technologies support these functions, their efficiency can be significantly improved with network assistance. Network assistance also provides session continuity by moving D2D traffic sessions back onto the cellular infrastructure network when/if the D2D link fails. To manage all of these assistance features, we introduce a new entity into the operator's core network called the *D2D server*. This server stores the subscribers' UE identifiers along with any P2P application IDs the users have provided. The server communicates with other entities in the operator's core network for the purposes of UE location estimation and with UEs to assist with device discovery, D2D connection establishment, and session continuity.

The D2D server also communicates with 3rd party P2P application servers for the purpose of updating user profiles with their associated D2D server names. P2P application servers maintain a database of users' P2P application IDs and offered P2P content and/or services. Registered users can access these P2P application servers to search for peers who are offering their desired P2P content/service. The system architecture of our proposed solution is illustrated in Figure 1 and works as follows:

(1) Each UE, upon user's command, uses the application-layer credentials to authenticate itself with the P2P application server (e.g. Facebook). This allows it to perform operations with content as well as authorize 3rd party access.

(2) The UE also authorizes its D2D server to update the user's profile on the P2P application server with the D2D server name (necessary information for any user that wants to connect via D2D with another peer). This process also verifies that the UE indeed belongs to the owner of the associated P2P application ID.

(3) The user may publish or search for the locations of P2P application content and/or services on the P2P application server.

(4) The UE asks the D2D server to assist in establishing a D2D connection, by resolving the application-layer content location provided by the P2P application server into an actual link-layer connection and IP address to which sockets can be bound.

(5) Finally, the P2P data exchange begins. Note that the P2P application server is not involved at this point. The D2D server, however, may monitor and adjust the properties of the D2D link as necessary. Interestingly enough, the devices participating in the data transfer do not require any prior authorization or direct contact. In addition, after the transfer is complete and D2D connection is terminated, the devices continue to be unaware of each other, even if they are just meters apart. This interesting detail allows us to envision anonymous sharing services, which are not possible with distributed systems, where devices give away their identity on regular basis by broadcasting discovery beacons.

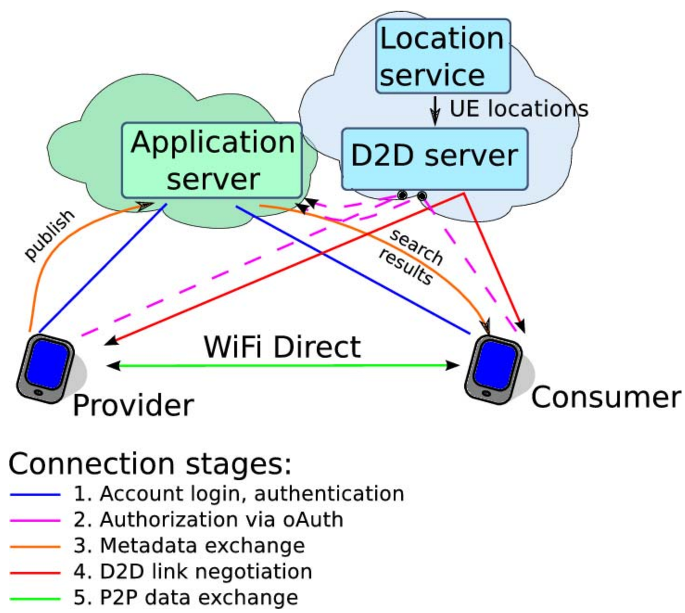

Figure 1. Assisted D2D connection establishment via the D2D server

The most important features of our solution are as follows:

– The solution upholds current security and permission models already employed by the application server. Users can restrict visibility of offered P2P content/services to certain peers. This means that malicious users that wish to retrieve restricted content would need to hack the P2P application server before they can even determine the location of their desired content, adding an extra layer of security. On the other hand users may share content to neighboring devices while remaining completely anonymous, never even exposing any of the permanent identifiers such as MAC addresses or lists of shared services.

– UEs do not have to broadcast discovery information or listen to discovery requests. Instead, they can keep their D2D radio interfaces off until the network informs them that they are in proximity of their desired peer. This is extremely important as active D2D radios tend to consume significant amounts of energy even if no data is being sent or received.

– Users can share P2P content/services without revealing their permanent device IDs. Since the network can manage and allocate temporary link layer IDs to its subscribers, the UEs never need to publish their permanent device IDs during discovery or connection establishment, providing perfect anonymity if needed.

– The cellular operator has the ability to monitor traffic and/or quality of service on network-assisted D2D connections. While the operator cannot directly monitor the traffic sent, it can still request periodic bitrate and/or QoS updates from the UEs. This information could be useful for network planning, as well as for coordination with existing infrastructure WiFi networks to mitigate interference.

One of the most notable advantages of network-assisted D2D is the ease of integration into the existing 3GPP LTE architecture. Figure 2 illustrates the system architecture of our solution. As mentioned previously, our solution requires only one new network entity, the D2D server, which will reside in the evolved packet core (EPC) of the network. This position allows the D2D server to communicate with the location center (SMLC) to learn the UE positions, while also allowing it to interact with the outside world application servers effectively.

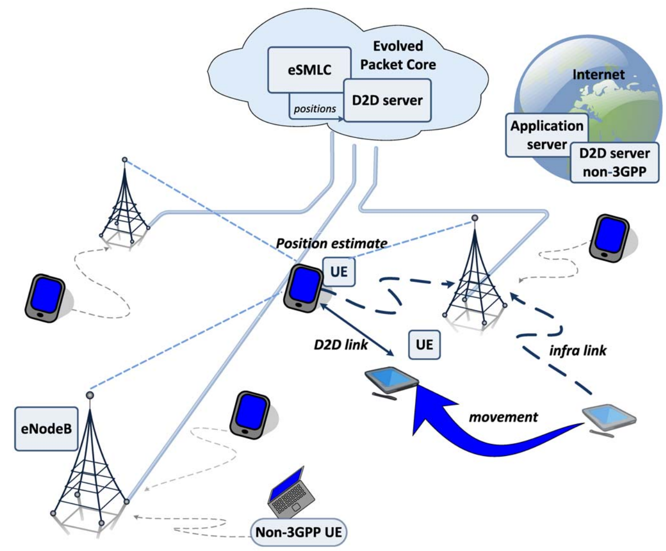

Figure 2. Proposed D2D services layout

III. Implementing traffic offloading prototype

In this section, we provide several key design details that make the implementation of our network-assisted D2D offloading system a reality within current web and Internet. As of today, neither of those are a part of any specification or standard, yet their simplicity serves as the proof of our concept.

A. Android and IP networks

Android as a Linux-based system[1] already allows to have simultaneous connections with more than one radio, yet if both 3GPP LTE and WiFi interfaces are active the UE has two directly connected networks and only one default gateway for sending its traffic outside of these networks. At this state, it is possible to reach the other peer on a D2D link only when the destination address of an IP packet is the WiFi Direct address of the peer and the source address is the one of the originating UE. Due to lack of spare IP addresses, however, WiFi Direct link uses private address range, which means that if the D2D link is ever disconnected the peer becomes unreachable, even if there is an alternative path present (because private range packets are not forwarded). For this reason, it is desirable to be able to reach peer's public IP address of 3GPP LTE interface through the WiFi Direct link.

One of our goals is thus to create a solution that would be transparent for already existing applications and this way ease the adoption of the proposed technology. For this reason, changes on the physical layer do not bring the desired results as it is heavily vendor-specific. The similar situation is with the link layer: putting rerouting logic into the existing hardware would be next to impossible, and creating virtual interfaces causes overheads. Since applications heavily rely on existing transport layer protocols, any modifications there are not possible either. On network layer, IP addresses are in a way bound to the physical interfaces, but the forwarding decisions are made independently of the interfaces. This allows us create an interface-independent solution without the need for modifications at the upper layers involving mobile IP/virtualization.

The default configuration of an Android system allows having multiple gateway routes, but gateways are inserted into routing table with different costs. This way no load-balancing is performed, and only one gateway route is used at a time. In the case of LTE (or any cellular) interface and WiFi, the LTE gateway route is preferred when the Internet connectivity is expected, whereas the WiFi link and especially WiFi Direct do not guarantee Internet connectivity at all. Hence, changing the cost of gateway route would make the Internet unreachable for all applications on the device.

We, therefore, propose here the *route injection* solution, which is based on injecting a custom route for a particular peer in the routing table. Owing to the Linux kernel in Android system, it is possible to enable routing by modifying the system value *net.ipv4.conf.all.forwarding* from 0 to 1. After this change, the mobile device has forwarding capabilities of

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

a conventional router, and thus will forward packets from one interface to another if necessary. It allows sending IP packets with the source address equal to LTE interface public IP address and the destination address equal to peer's WiFi Direct private IP address, and the IP layer of Android system will send them through a WiFi link.

Since we only want to offload communication with a single peer, we can insert a route into the routing table saying that the peer's LTE public IP address is reachable through his WiFi Direct private IP address. The insertion is performed by the command

```
ip route add PEER_LTE_IP/32 via PEER_WD_IP
```

Once done, all the traffic with destination IP address PEER_LTE_IP will be forwarded to peer's WiFi Direct interface, with the shortest path going through the WiFi link, as intended. When not needed anymore, the route can be removed by running

```
ip route del PEER_LTE_IP/32 via PEER_WD_IP
```

The insertion and/or removal is performed by the management app that is running as system service. Both operations are performed as commanded by D2D server, but the removal can also be based on link conditions (for instance, poor RSSI).

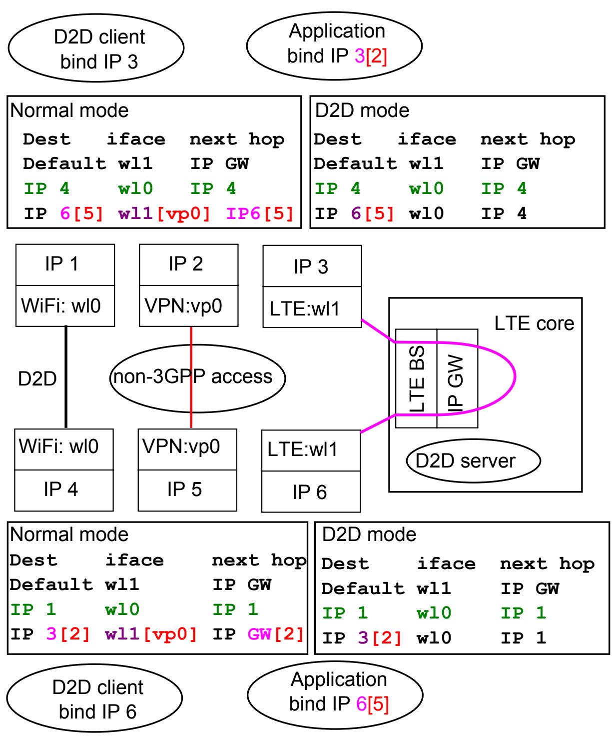

Figure 3. Route injection example

When reaching a particular signal-strength threshold, the route can be inserted or removed as needed. An example of the routing table during injection in case of WiFi Direct is presented in Figure 3. One may see that the traffic may be steered not only through WiFi Direct, but also through other forms of non-3GPP access, e.g. a campus networks.

The described routing table injection does not have to be performed on both devices participating in the D2D offloading. In fact, the return traffic can continue traveling through the cellular interface, which may be useful in some scenarios. Also the injection is not limited to a single peer: as many peers as necessary may have their own custom routes provided that they are all allocated different private IP addresses (D2D server can ensure this).

B. *Infrastructure and platform support*

Our technology prototype for network-assisted cellular traffic offloading relies on the three components:

a) Client-side application running on UEs;

b) Content register service;

c) ProSe D2D server.

The content register is assumed to be a web platform providing data sharing services to its users, e.g. social networks like Facebook, Google+, etc. The D2D server is designed to be run by the network operator or Internet service provider (ISP). It is worth noting that alternatively to the suggested architecture layout, our service is flexible enough for both entities to be run by a single authority or even on the same host.

For the purposes of our D2D technology demonstration, both server side applications were deployed within our cloud infrastructure. Two virtual machines were setup to act as the content register and the D2D server. Content register part was implemented in PHP served by Apache web server. Thus the application is a regular website that gives a registered user the possibility to post its intent to share some content, or to search for shared data records (in essence acting as a torrent tracker). The user posts only the information required to access the data rather than its location, i.e. sharing the protocol and port number, while the IP address that locates the data is handled by the D2D server. To identify the new addressing scheme required by D2D connections, we have introduced a new protocol identifier in the URI returned by the content register: *"d2d://"*. End user devices were configured to interpret this protocol type as a request to start Client application, that in turn is capable of communicating with the D2D server to resolve the username of the data serving peer into an actual connection to that peer.

D2D server was implemented in Python under Twisted framework as a standalone application, and utilizes HTTPS as transport for the control messages. The system assumes the UE's mobile data link to be up during the service usage. Our implementation shows that content register and D2D server(s) are easily integrated in a seamless fashion into the existing web serving infrastructure, and do not require any obscure design tricks.

UE's employed by our technology demonstration are Sony Xperia ZL phones, provided by Sony Mobile. Android, as an open source mobile platform provides the needed flexibility in configuration and available tools to fulfill the requirements for the user devices. One of the main features required is that the UEs should have both cellular and WiFi connections up simultaneously. Due to the energy consumption, most of the devices on the market restrict their network connectivity to only one interface at a time. To enable both interfaces simultaneously,

	loaded HSPA or EDGE	conventional HSPA+/DC-HSPA	LTE (empty cell)	Assisted-WiFi, 10m	Assisted-WiFi, 20m	Plain WiFi Direct, 5m
Sustainable rate	10-70kbps*	300-500 kbps	3Mbps	5Mbps	500-600 kbps	5 Mbps
Stream latency	timeout	300-500 ms	200-300 ms	<50 ms	100-200 ms	<50 ms
Signaling latency	1-60 s*	1-2 s	400-500 ms	N/A	N/A	150 ms
Activation delay	5-100 s*	3-5 s	~1 s	500-900 ms	1-3 s	700-800 ms

Table I. COMPARATIVE PERFORMANCE OF OFFLOADING SYSTEM

we had to bypass the native Android service controlling WiFi (this can be achieved by disabling the corresponding service), and interact with the WiFi driver directly. As Android system is derived from GNU Linux, it provides us with the needed tools: *wpa_supplicant* interface controlled via *wpa_cli* utility.

Another issue is that in order to install these GNU utilities, the Android OS has to be compiled with user debugging enabled, and the stock firmware provided by most manufacturers does not allow that. The final choice for the user device software platform was the aftermarket firmware Cyanogenmod maintained by the FreeXperia group (http://www.cyanogenmod.org), which can be deployed by anyone who owns a supported UE. To summarize, the changes to the UE's include root access, a service to communicate with D2D servers, and deployment of tools to control the WiFi interface.

Another requirement for the UE is to be capable of accept connections from the cellular data link. Considering the fact that most operators use the private IPv4 address pool to assign to the user devices, and provide Internet connectivity through cone NAT, the access to the services running on the user's device from outside its local link is not trivial. One of the possible solutions to overcome this issue would be using the IPv6 addresses, but the mobile operators do not rush towards IPv6 support, as replacing the existing infrastructure is costly. The simplest tested option for our technology demonstration was encapsulating the 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. However, due to excessive complication of the communication and large tunnel overhead, this approach is not very scalable. Discussing this issue with local operators, we were able to negotiate with TeliaSonera Finland Oyj for a certain access point that would provide the user devices with public routable IPv4 address. Later, similar agreement was reached with AT&T in the US.

IV. RESULTS OF QOE EVALUATION

The evaluation of our D2D technology prototype was performed in two directions. First, system-level simulations have been used to make sure the solution is scalable in practical network deployments. In the course of this study, we have developed an advanced system-level simulator (SLS) based on up-to-date LTE evaluation methodology and current IEEE 802.11 specifications. This SLS is a flexible tool designed to support diverse deployment strategies, traffic models, channel characteristics, and wireless protocols. It models all of the conventional LTE/WiFi Direct infrastructure and client deployment choices (hexagonal vs. square cells, environment with or without wraparound, uniform vs. clustered client distribution, etc.). With the help from our SLS, we demonstrated how network-assisted D2D communications allow users to benefit from higher data rates over shorter distances without compromising their battery life. More details about this work can be found in our recent publications [13], [14]. To summarize, our results indicate that the capacity gains with massive WiFi Direct D2D deployments could typically reach as much as 50% of the original cell uplink capacity, yet of course in some cases when the D2D peers are very close to each other the gains could be even higher.

The second evaluation direction has been to assess the actual end-user quality of experience while using our D2D prototype under different conditions. This was done with the live Sony Xperia devices and various multimedia applications as benchmarks. The performance of the video streaming as well as that of the connection management procedures has been thus evaluated.

A. Testing video on demand

The video on demand tests have been based on the idea that a user is likely to be sharing a short clip (e.g. YouTube video), which is popular in some area. The video clips for the testing have been selected with various bitrates starting from 300 kbit/s (very poor quality) and up to 5 Mbit/s (HD quality). The duration of the clips was chosen to be around 5 minutes. The networks used for testing were TeliaSonera Finland (LTE, DC-HSPA), AT&T US (HSPA+, HSPA), and T-Mobile US (LTE, DC-HSPA, HSPA). The offloading test was performed under different conditions of cellular networks, with the results presented shortly in Table I.

In Table I, * indicates timeouts, or conditions under which the service can be deemed unusable. Signaling delay is the time needed for the service messages to take effect, while offloading delay is the time it takes for interface to go online. D2D signaling latency is significantly higher than streaming latency due to the fact that multiple messages need to be exchanged before commands are executed by UE's. Relative to WiFi Direct, the proposed solution shows similar performance delay-wise, but the reduced power consumption makes it vastly superior. One may argue that the table does not cover all the cases, yet it indicates the potential bottlenecks in network-assisted offloading.

Further, when it comes to WiFi Direct, one would expect interference to play a major role resulting in poor indoor performance [19]. We have tested D2D in office environments with on university and large company campus networks, in open-air with minimal interference, and in urban environment, with approximately same results. From the user's point of view, as far as video streaming is concerned, even in a busy office the link length has a much stronger effect on the transmission quality than interference, with limiting distance of approximately 50 meters. We have not observed such WiFi congestion that HD video streaming would not be possible, even on university campus with massive amounts of interfering access points on all frequencies.

Our summary is that depending on the quality of the cellular link the usefulness of the D2D connections may vary.

However, even with the state-of-the-art LTE technology, the MAC transfer delays are up to 10 times higher than those with WiFi, and with 3G cellular technologies HD video streaming is not even possible due to capacity limits of real deployments.

V. Conclusions

For mobile network operators, D2D connectivity is becoming vital to enable traffic offloading from the core network and to realize efficient support of social networking through localization. Along these lines, our network-assisted D2D technology prototype has been implemented to identify the major challenges and potential gains of enabling direct connectivity between proximate mobile devices. As our previous research has already confirmed significant capacity gains on the system level, the user QoS experience has become the main focus of the current prototype implementation. Below we summarize our most important findings and lessons learned.

1. The successful integration of the D2D connectivity with the existing web application paradigm shows that there are no significant problems that would prevent the application service providers from enabling D2D communications for their clients. Moreover, some of them could benefit significantly by using this new infrastructure to design new services that were not possible before without continuous GPS tracking.

2. Successful deployment of the network-assisted D2D service into the Android platform indicates that the OEMs will not face any significant challenges either while implementing the corresponding management protocols. Certain platforms, especially those that do not fully comply with Internet standards, may however face certain difficulties with the route injection procedure required to steer traffic through the D2D link once it is established, but that would primarily become a concern of the manufacturer, not the users.

3. The quality of running D2D links by far exceeds the best cellular connections within reasonable distances, e.g. below 50 meters. WiFi Direct links enable HD video streaming as well as real-time applications easily and with good energy efficiency.

4. Sophisticated firewalls deployed by mobile operators complicate the D2D connections on cellular, for as long as IPv4 remains the predominant addressing solution, as there are just more mobile devices out there than are there IPv4 addresses.

5. The success of the entire proximal D2D concept relies heavily on the operator's support for cellular positioning, as well as some cooperation between the operators. This should become reality once the appropriate standards are completed by 3GPP.

Overall, we are confident that the challenges identified during the implementation of our D2D offloading prototype will be resolved within the next year or two, with first services supporting the novel features following shortly after. As the pressure from both the services and the capacity points of view is rising, it is only a matter of time before market solutions are deployed, and our scalable and easy-to-implement technology prototype detailed in this work is a decisive step forward. We also believe, that with deployment of D2D servers, more intelligent ways to manage D2D connections in a centralized fashion will be found, with potential gains to capacity and efficiency.

Acknowledgment

This work is supported by Intel Corporation, TISE, and the IoT SRA program of Digile, funded by Tekes. The team would also like to thank Sony Mobile for supplying the hardware required for the demo. Last but not least, we acknowledge the help received from mobile operators, TeliaSonera (Finland) and AT&T (US), kindly assisting us with evaluations.

References

[1] *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012-2017p*, 2013.

[2] C. Sankaran, "Data Offloading Techniques in 3GPP Rel-10 Networks: a Tutorial," *IEEE Communications Magazine*, vol. 50, no. 6, pp. 46–53, 2012.

[3] Ericsson white paper, "More than 50 billion connected devices," tech. rep., 2011.

[4] D. Raychaudhuri and N. B. Mandayam, "Frontiers of Wireless and Mobile Communications," *Proceedings of the IEEE*, vol. 100, pp. 824–840, Apr. 2012.

[5] A. Marques, L. Lopez-Ramos, G. Giannakis, J. Ramos, and A. Caamano, "Optimal Cross-Layer Resource Allocation in Cellular Networks Using Channel and Queue-State Information," *IEEE Transactions on Vehicular Technology*, vol. 61, pp. 2789–2807, July 2012.

[6] K. Doppler, A. Osseiran, M. Wodczak, and P. Rost, "On the Integration of Cooperative Relaying into the WINNER System Concept," in *2007 16th IST Mobile and Wireless Communications Summit*, pp. 1–5, IEEE, July 2007.

[7] A. Pyattaev, S. Andreev, O. Galinina, and Y. Koucheryavy, "System-Level Evaluation of Opportunistic Client Cooperation in Wireless Cellular Networks," in *Proceedings of 20th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–6, IEEE, July 2011.

[8] *3GPP LTE Release 10 & beyond (LTE-Advanced).*

[9] X. Wu, S. Tavildar, S. Shakkottai, T. Richardson, J. Li, R. Laroia, A. Jovicic, 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 *Forty-Eighth Annual Allerton Conference*, pp. 514–521, 2010.

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

[11] K. Doppler, C. B. Ribeiro, and J. Kneckt, "Advances in D2D communications: Energy efficient service and device discovery radio," in *2nd International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology (Wireless VITAE)*, 2011.

[12] 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.

[13] A. Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy, "Proximity-Based Data Offloading via Network Assisted Device-to-Device Communications," in *IEEE Vehicular Technology Conference*, 2013.

[14] A. Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy, "3GPP LTE Traffic Offloading onto WiFi Direct," in *IEEE WCNC Workshops*, 2013.

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

[16] 3GPP, "TR 23.890 Optimized Offloading to WLAN in 3GPP-RAT mobility," tech. rep., 3GPP, 2012.

[17] 3GPP, *SP-110638 - LTE Proximity-Based Services Study Item.* 2012.

[18] J. Andrews, "Can cellular networks handle 1000x the data?," tech. rep., UT Austin, 2011.

[19] P. Ali-Rantala, L. Ukkonen, L. Sydanheimo, M. Keskilammi, and M. Kivikoski, "Different kinds of walls and their effect on the attenuation of radiowaves indoors," in *Antennas and Propagation Society International Symposium*, pp. 1020,1023, IEEE, 2003.