Softwarized LTE Self-Backhauling Solution and Its Evaluation

José Gamboa, Ilker Demirkol
Department of Network Engineering
Universitat Politecnica de Catalunya, Barcelona, Spain
Email: {jose.umberto.gamboa, ilker.demirkol}@entel.upc.edu

***Abstract*—Long Term Evolution (LTE) is projected to be the mostly used mobile technology before 2020, thanks to the high data rates, all IP-based and simplified architecture it brings compared to previous mobile network technologies. To address the exponential data requirements of the future, a targeted solution for LTE is network densification, i.e., deploying small cells (cells with reduced radius compared to macro cells), which will enable better frequency reuse. However, a challenge here is the backhauling of the evolved-NodeBs (eNBs) used for small cells, since the fiber-based backhauling is costly and is not always feasible given the physical environment. For this, a recent idea is to use LTE self-backhauling, where an eNB can relay its data to another eNB through the use of LTE technology. In this paper, we develop and evaluate an implementation of LTE Self-Backhauling building on an open-source software and commodity hardware (specifically, regular PCs and low-cost software-defined radios) for the LTE system. For this, we implement a self-backhauled eNodeB, which connects to another eNB through an LTE connection. Through physical experimentation using off-the-shelf user equipments (UEs), we show that the method proposed is viable and can improve the network coverage and the throughput of the network.**

***Index Terms*—LTE, 4G/5G mobile networks, Donor eNB, LTE Relaying, Relay Node, Self-Backhauling.**

I. Introduction

In recent years, LTE has been the most deployed broadband communication technology. Although there is rigorous effort for the 5G standardization, LTE is expected to be the dominant mobile technology for the next decade. To address the projected exponential increase in mobile traffic demand and the new application types that require better coverage (e.g., IoT), operators are looking into ways to increase the capacity and the radio coverage of their LTE networks. One common solution targeted is the cell densification, which increases frequency reuse, and hence the capacity of the network.

LTE, by design, considers fixed base stations, which are physically connected to the core network through a transport network. Such connection is known as backhauling, and wired backhauling solutions (e.g., fiber) are commonly used for cell backhauling. However, the cell densification brings a significant challenge to this conventional solution, due to the installation and maintenance costs of the wired solutions. Moreover, there are locations where, for example, a fiber connection is not possible due to the physical environment.

An alternative wireless backhauling solution to these challenges is *self-backhauling* [1]. Self-backhauling enables the wireless backhauling of an eNB (also named as Relay Node (RN)) using the existing LTE radio interface of another eNB (named as Donor eNodeB (DeNB)) as a backhaul link.

Using the existing LTE radio interface for backhauling provides a better cost-efficiency solution, since in this case the backhaul and access share the same Operation and Maintenance (O&M) system, simplifying the system management, and achieves a higher spectrum utilization, thanks to the potential reuse of time, frequency and space resources between access and backhaul link [1].

Although 3GPP included LTE self-backhauling in Release 10, its uptake has been limited. However, due to the densification targets of the operators (e.g., through small cell deployments), there is a fresh interest in wireless backhauling solutions (e.g., between small cells and macro cells [9]). In addition, operators also look for softwarized networking solutions to achieve flexibility, to avoid vendor lock-in, and hence to reduce both CAPEX and OPEX of their networking solutions.

To achieve these targets, we propose an LTE self-backhauling solution that is built using: i) open-source LTE software project that provides the eNB and Evolved Packet Core (EPC) software, and ii) commodity hardware such as PCs and software-defined radios (SDRs). Hence, our implementation provides a low-cost and flexible wireless backhauling solution. In this paper, we first present our solution and then evaluate it through physical experiments to assess its viability and the performance improvements it achieves in terms of coverage and the capacity of the network. An implementation challenge has been the double GPRS Tunneling Protocol (GTP) encapsulation problem encountered (due to the self-backhauling), which is solved through a virtualized EPC solution as detailed in this paper. Experiment results show that our solution is viable and increases the coverage and the capacity of the network as expected.

Within the limited related work from the literature, we can highlight the enhanced UEs (eUEs) proposed in [4] as active network elements to relay eNB traffic, and therefore achieve LTE self-backhauling. The eUEs are able to forward L2/MAC packets, for which a low latency communication is targeted through the use of a buffer-aware scheduling. However, such solution requires significant changes at UE, and the incentive and cooperation mechanism for battery-powered UEs to relay traffic for eNBs are key challenges for its viability.

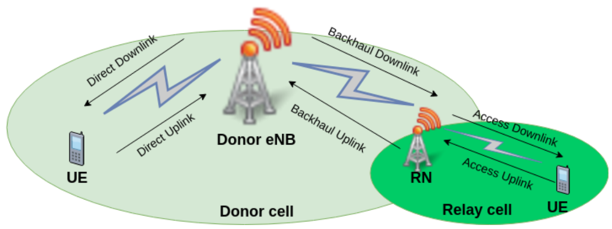


Fig. 1. Relay Network

The concept of enhanced evolved NodeB (e2NB) is introduced in [5], which targets a wireless mesh backhaul link between e2NBs, using the existing LTE air interface. The e2NB implementation combines the conventional eNB with several virtualized UEs (running on the eNB machine), each communicating with a neighbor eNB. Mobility Management Entity (MME) and Home Subscriber Server (HSS) components of EPC is also included in e2NB to allow standalone functioning. The developed solution is experimented with an LTE emulator/simulator, showing its feasibility. Nevertheless, the proposed solution requires significant modification at eNBs, calls for a copy of HSS to individual eNBs (with the UEs that can connect), and no physical experimentation evaluations have been performed.

To the best of our knowledge, this paper presents the first study in the literature on implementation and physical evaluation of an open-source LTE self-backhauling solution using commodity hardware.

II. LTE Self-Backhauling Concept and 3GPP Specification

In order to improve and extend the radio coverage area or increase the capacity of 4G networks, the concept of *relaying* was defined for LTE-Advanced in 3GPP Release 10 of the LTE specifications [7] [8]. Fig. 1 illustrates the concept, which includes the following terminology used in 3GPP for a Relay Network. *Relay Node (RN)* is a low power base station that relays UE traffic wirelessly, and *Donor eNB (DeNB)* is the anchor eNB for the RN. *Backhaul link* is the one established between the DeNB and the RN, and *Access link* is the one between the RN and the UE. There might also be UEs with a *Direct link* to the DeNB. The RN is connected to the DeNB using its existing LTE radio interface.

Relaying strategies can be categorized by the protocol layer functionality of the RN [10]. In 3GPP Release 8, a relay as a form of repeater was introduced, which is known as wireless repeater or Layer 1 Relay Node (L1 RN) [6]. This repeater receives the signal of the eNB, then amplifies and re-transmits it. The process of the repeater is performed at Layer 1 (i.e., at physical layer (PHY)). L1 relay is a good solution for mitigating coverage holes. It is cost effective and has low latency compared to the other relay types. However, as it amplifies the signal coming from the DeNB, it also amplifies the noise and the interference, and transmit them together to the destination along with the desired signal.

In Layer 2 Relay Node solution, RN forwards user plane and control plane traffic in the sublayers Packet Data Convergence Control (PDCP), Radio Link Control (RLC) and Medium Access Layer (MAC). Here delay is introduced, since the relay decodes and re-encodes the received data, but there is no noise added to the data signal that is forwarded by the L2 RN. L3 RN, is similar to the L2 RN, but L3 RN also forwards IP packets and supports Radio Resource Control (RRC) functionalities such as mobility between RNs. A solution based on L3 RN called self-backhauling was proposed by Hoymman et al. in 2008 [2]. Such self-backhauling relay has the same functionality as an eNB, but it can transmit a lower power, where it has a smaller cell size than the eNB. The relay must support LTE radio interface protocols, since it is connected to the eNB through LTE radio interface for communication. Such solution is also assessed towards 5G in [1].

If the access and backhaul links share the same carrier frequency, then the RN is named as inband RN (outband RN, otherwise). If the RN can communicate both on access and backhaul links at the same time (achieved through antenna isolation, etc.), it is called full-duplex RN (otherwise, half-duplex RN). In relevant 3GPP discussions, four RN types were proposed based on these definitions. Type 1, 1a and 1b relays correspond to L3 RN. They have their own physical cell identity and appear as a regular eNB to UEs. Type 1 RNs are defined to be inband half-duplex, whereas Type 1b RNs are inband full-duplex, and Type 1a RNs are outband full-duplex. Type 2 relays correspond to L2 RN, i.e., do not have their own physical cell identity, nor control channels and are transparent to the UE. Although 3GPP LTE-A specifications included only Type 1 and 1a RNs in the relevant specifications, Type 1b RNs can be deployed as specific RN implementation. In this paper, we implement and evaluate both Type 1a and 1b RNs for the purpose of LTE self-backhauling solution.

III. Implementation

For comparative purposes, two LTE network implementations were carried out: 1) a conventional LTE network setup that includes an eNB and relevant EPC components establishing only a direct link with a UE, and 2) the self-backhauling network setup including an RN, a DeNB, a UE, and relevant EPC components, establishing both access and backhaul links.

A. Conventional LTE Architecture Setup

To implement an open-source and commodity hardware based LTE setup, we use Open Air Interface (OAI) software suite [3], which includes eNB and EPC (specifically MME, HSS, SGW (Serving Gateway) and PGW (PDN Gateway)) software. For eNB execution, an SDR is needed, otherwise the rest of the components are based on software running on PCs. The conventional LTE network scenario implemented corresponds to the setup shown in Fig. 2a. In the following, the specific configuration and setup details of OAI components are presented.

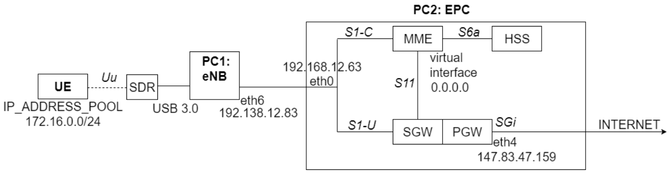


Fig. 2. The conventional LTE network setup configuration.

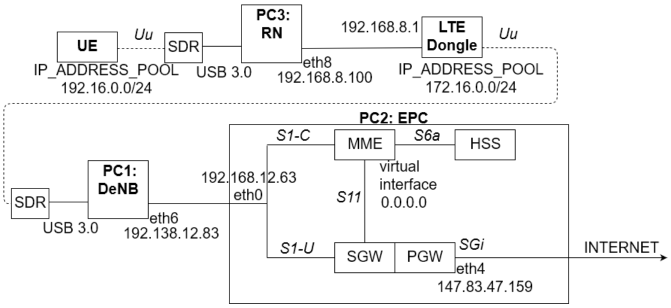


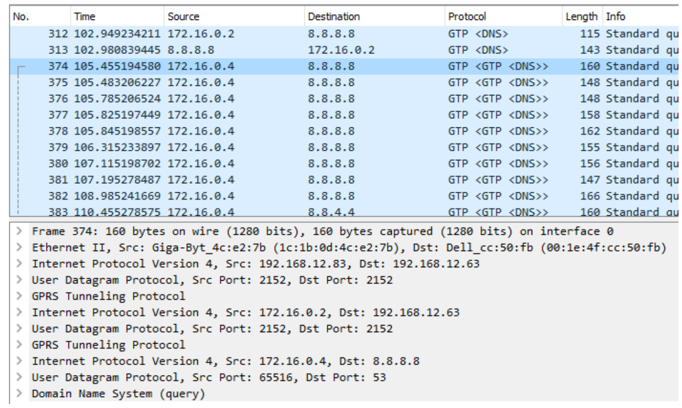
Fig. 3. LTE Self-Backhauling network setup configuration.

For conventional LTE architecture setup, as shown in Fig. 2, we use PC1 for eNB and PC2 for EPC, which are connected through an Ethernet cable, and the connection between PC1 and USRP B200 is via USB 3.0 for the high data traffic between the two. Baseband processing is done on PC, which requires a real-time operating system. Low-latency Linux kernel version 3.19 was used for this purpose. As shown in the figure, all the EPC components run on the same machine, which has two Ethernet interfaces, one for eNB and one for Internet (i.e., SGi) connection.

B. Self-Backhauling LTE Architecture Setup

The implemented network configuration for the self-backhauling LTE setup is shown in Fig. 3. In the self-backhauling architecture, the implementation of DeNB has been similar to the conventional LTE architecture setup, i.e., a low cost USRP B200 (Universal Software Defined Radio) is used and connected to a PC (PC1) that runs the software for DeNB, which is connected via Ethernet to PC2 that executes EPC VNFs. For the Self-Backhauled network, a second USRP B200 is used for access link and connected to a PC3, which works as RN. To achieve the full-duplex communication and for ease of implementation, the RN uses an LTE dongle for backhaul link. This allows the RN to connect to DeNB as a UE, as specified in 3GPP specifications. LTE dongle used (Huawei E3772) creates an automatic Ethernet interface and assigns a specific address pair to this interface (one for dongle itself and one for the PC it is connected to).

Note that a softwarized UE solution at RN can also be used, e.g., OAI UE or srsUE, however, this would require the use of another SDR at RN, which would increase the monetary cost of this solution and the computational complexity at the RN (as both eNB and UE require real-time operation).



No.	Time	Source	Destination	Protocol	Length	Info
312	102.949234211	172.16.0.2	8.8.8.8	GTP <DNS>	115	Standard qu
313	102.980839445	8.8.8.8	172.16.0.2	GTP <DNS>	143	Standard qu
374	105.455194580	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	160	Standard qu
375	105.483206227	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	148	Standard qu
376	105.785206524	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	148	Standard qu
377	105.825197449	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	158	Standard qu
378	105.845198557	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	162	Standard qu
379	106.315233897	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	155	Standard qu
380	107.115198702	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	156	Standard qu
381	107.195278487	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	147	Standard qu
382	108.985241669	172.16.0.4	8.8.8.8	GTP <GTP <DNS>>	166	Standard qu
383	110.455278575	172.16.0.4	8.8.4.4	GTP <GTP <DNS>>	160	Standard qu

> Frame 374: 160 bytes on wire (1280 bits), 160 bytes captured (1280 bits) on interface 0
> Ethernet II, Src: Giga-Byt_4c:e2:7b (1c:1b:0d:4c:e2:7b), Dst: Dell_cc:50:fb (00:1e:4f:cc:50:fb)
> Internet Protocol Version 4, Src: 192.168.12.83, Dst: 192.168.12.63
> User Datagram Protocol, Src Port: 2152, Dst Port: 2152
> GPRS Tunneling Protocol
> Internet Protocol Version 4, Src: 172.16.0.2, Dst: 192.168.12.63
> User Datagram Protocol, Src Port: 2152, Dst Port: 2152
> GPRS Tunneling Protocol
> Internet Protocol Version 4, Src: 172.16.0.4, Dst: 8.8.8.8
> User Datagram Protocol, Src Port: 65516, Dst Port: 53
> Domain Name System (query)

Fig. 4. Wireshark capture of incoming packets in S-GW

C. Double GTP Problem

After compiling and running the LTE self-backhauling setup, a problem was found that prevented the UE to connect to the Internet. After detailed investigations, it was found that since RN does the GTP encapsulation of the data received from UE to send the data to DeNB (given that Type 1a/1b RN acts as a standalone eNB), and since the DeNB does not perform any specific processing for the packets coming from the RN, DeNB was realizing another GTP encapsulation on top of the incoming packets from RN. This resulted in double GTP encapsulation which is shown in the wireshark captures in Fig. 4.

Fig. 4 shows the packets that DeNB sends to the S-GW, which in turn use GTP decapsulation once to send these packets to the P-GW. This resulted in having GTP encapsulation in the packets sent to the Internet by P-GW.

The possible solutions we foresee to mitigate the problem of double GTP encapsulation are *1) Update the eNB code at DeNB* to differentiate packets from a UE and RN. If the packets are from RN, it will forward them without any GTP encapsulation to the EPC. This requires a specific information collection at DeNB about the RNs connected (or might get connected for mobile RNs), which requires a complex handshake with other EPC or network management entities. *2) Update the SGW code* to be able to check for GTP encapsulation towards itself recursively. *3) Use an SGW at RN* to remove the GTP encapsulation before sending the packets to the DeNB.

In this paper, we focus on the last option, which does not require any deviation from LTE specifications, easy to implement and can support RN mobility. The resulting setup at RN is shown in Fig. 5. Since in OAI implementation, SGW and PGW are one executable, namely SPGW, and one MME can associate to one SPGW, a virtual MME and a virtual SPGW (respectively, vMME and vSPGW) were added to the RN PC. Hence, the eNB software of RN communicates with vMME and vSPGW running in the virtual machine (VM) through a virtualized interface (shown as vmnet8). Moreover, all these three executables communicate with the LTE dongle

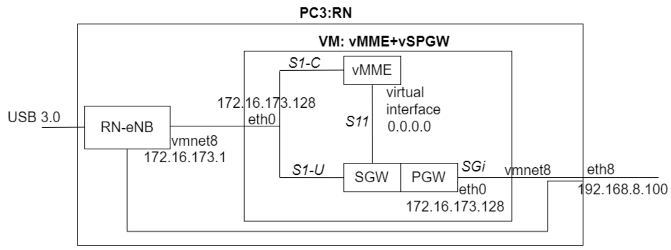


Fig. 5. LTE Self-Backhauling setup at RN with vMME and vSPGW.

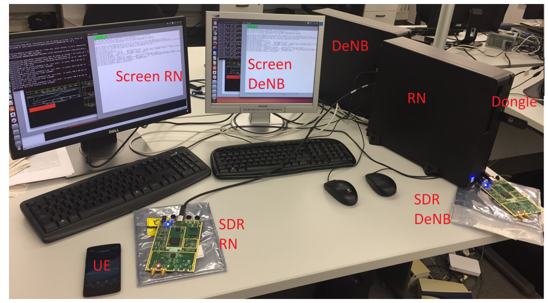


Fig. 6. Self-Backhauling LTE network implemented in a real environment

through the Ethernet interface defined by it. The rest of the network setup is the same with that of Fig. 3.

IV. Evaluation

In order to evaluate the implemented setups, we use LTE Band 7, i.e., the central frequencies between 2620 and 2690 MHz. We compare the network performances achieved by a UE connected to a conventional LTE network, i.e., having a direct link with DeNB, and the same UE connected to the same DeNB through an RN, which uses LTE self-backhauling. The throughput is measured in terms of uplink and downlink bitrate using iperf and speedtest Android applications, respectively. The reason we use the speedtest Android application for measuring downlink bitrate is that, in the LTE self-backhauling setup, the UE is connected to the vMME and vSPGW of the RN and not to the MME and SPGW of the conventional LTE network. The vSPGW applies Network Address Translation (NAT) for the UE and sends the traffic with the IP of the dongle LTE given by the EPC. Hence, the SPGW that DeNB connected to is not aware of the IP of the final UE, which prevents a downlink connection from that SPGW to the UE. For the iperf tests, we evaluated TCP connections between the UE and the PC2. Multiple tests (around 20) were run to evaluate the variations in the performance. Since the variation between different runs is found to be insignificant, only the average performance results are presented in this section.

We used 2660 MHz as a carrier frequency for conventional LTE network and for inband (Type 1b RN) self-backhauling network evaluations, and for outband (Type 1a RN) self-backhauling network we used 2640 MHz and 2670 MHz as carrier frequencies for RN and DeNB, respectively. We vary the physical resources blocks by changing the LTE bandwidth between 5 MHz (25 PRB), 10 MHz (50 PRB), and 20 MHz (100 PRB).

Fig. 6 shows the implemented setup using the commodity hardware explained in the previous section. As seen in Fig. 6, there are two PCs, one for DeNB and the other for RN. In the RN there is an LTE dongle connected to the USB port. As explained before, this will allow RN to work also as a UE. There are two SDRs, we can see in the figure, one is for DeNB and the other for RN, and finally the UE that is close to the SDR of the RN is the one used for the test in this study.

The layout seen in Fig. 6 is inline with small cell self-backhauling scenarios, where the small cell is physically close

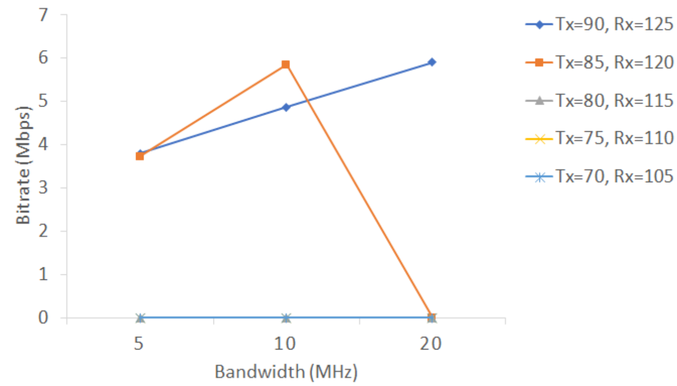


Fig. 7. Uplink throughput of UE for conventional LTE network setup.

to the UE, and through directional antennas it can achieve high gains in its backhaul link (imitated by close proximity of LTE dongle to DeNB SDR). We vary the "Tx" and "Rx" gain parameters of DeNB to change the size of its cell, and evaluate the conventional LTE network setup under different gain values. The default Tx and Rx gain values provided by the OAI implementation are 90 and 125 dB[1], respectively. For the evaluations, we fix the UE location and evaluate the throughput results of the network for UL and DL for varying gain value tuples.

V. Comparative Performance Results

The throughput values achieved for conventional LTE network setup are shown in Figs. 7 and 8 for uplink and downlink, respectively. As seen in the figures, the UE has non-zero throughput values when (Tx, Rx) gain tuple is (90dB, 125dB) for any bandwidth values, and when it is (85dB, 120dB) for 5MHz and 10MHz bandwidth. The rest of the throughput results are zero, which means the UE can not access the LTE network, i.e., it is out of the Donor cell. As expected, higher gain values enable better throughput performance.

For LTE self-backhauling evaluations, we vary the (Tx,Rx) gain tuple for the DeNB, while fixing it for RN at (75dB, 110dB) to generate a small cell size. We first evaluate the Type 1b RN solution, which is the inband RN solution with

[1] Note that the actual Rx gain passed to USRP is computed by OAI software by subtracting a calculated calibration offset.

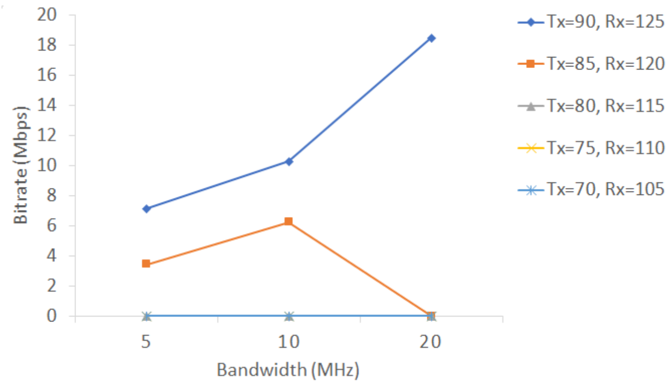


Fig. 8. Downlink throughput of UE for conventional LTE network setup.

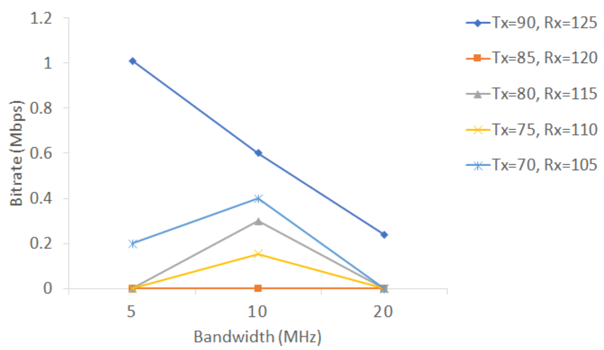


Fig. 9. Uplink throughput of UE with Type 1b RN using LTE Self-Backhauling.

full-duplex connection. Figs. 9 and 10 show the throughput results of this LTE self-backhauling setup.

As seen in Fig. 9, the UL throughput is very low, which happens because of the self-interference that exists when an inband solution is used. Looking at the downlink performance for Type 1b RN (Fig. 10), we see that we obtain results for all the tested values of (Tx,Rx) gain tuples, in contrast to Fig. 8. Hence, with a self-backhauling solution we can extend the coverage of the conventional LTE network, even if an inband solution is used.

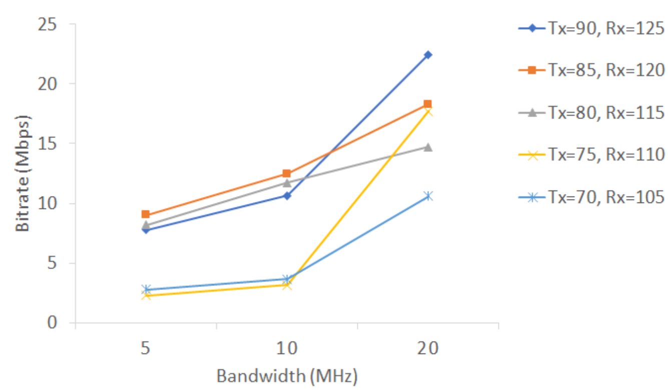


Fig. 10. Downlink throughput of UE with Type 1b RN using LTE Self-Backhauling.

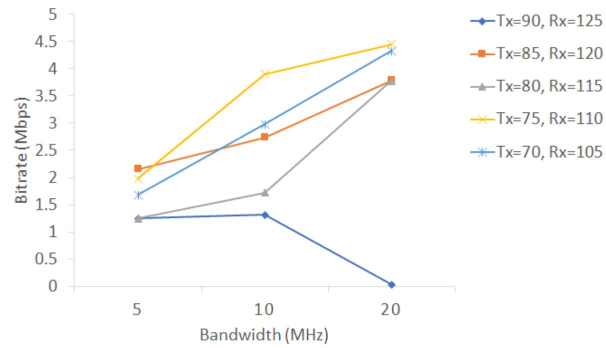


Fig. 11. Uplink throughput of UE with Type 1a RN using LTE Self-Backhauling

Note that although full-duplex implementation is done at RN (separate transceivers for backhaul and access links, and physical separation of antennas), there can still be self-interference in our inband setup due to non-ideal antenna types (omnidirectional). Yet, we observe performance improvement in downlink bitrate and not in uplink bitrate compared to conventional LTE setup in the inband solution. The reason can be explained by the type of modulation they use, since downlink uses Orthogonal Frequency Division Multiple-Access (OFDMA) and UE uplink uses Single-Carrier Frequency-Division Multiple-Access (SC-FDMA). OFDMA uses multiple subcarries, which can mitigate better interference than the SC-FDMA which uses only a single carrier. These performance results would improve significantly in a real setup with more idealistic antenna setups.

TABLE I
CQI LEVELS FOR INBAND SELF-BACKHAULING AT DIFFERENT TX GAINS

Bandwitdh (MHz)	**DeNB Gain (dB)**					**RN Gain (dB)**
	70	**75**	**80**	**85**	**90**	**75**
5	9	10	12	14	14	13
10	8	10	10	12	14	11
20	6	8	9	11	13	10

In Table I, the CQI levels (i.e., the downlink channel quality) for DeNB and RN are shown for Type 1b setup. As seen in Table I, the CQI level improves as the Tx gain value is increased for DeNB. The evaluations showed that the RN CQI does not vary with the DeNB Tx gain value, which shows the channel capture effect of the RN for its access link to UE. These results explain the improvement in downlink throughput, as the interference of backhaul link to the access link is not significant for downlink, and hence an inband solution is feasible. Comparing the CQI values from Table I to the maximum CQI value of 10 measured in the conventional network setup for the direct link, we can clearly see that the LTE self-backhauling improves the coverage. However, as mentioned before, uplink interference is a problem in LTE self-backhauling due to the limited transceiver complexity at UEs (both for UE itself and for the LTE dongle in our setup) for the transmissions, and requires careful antenna isolation.

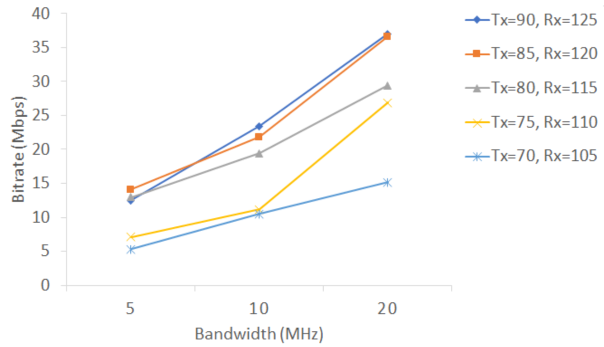


Fig. 12. Downlink throughput of UE with Type 1a RN using LTE Self-Backhauling.

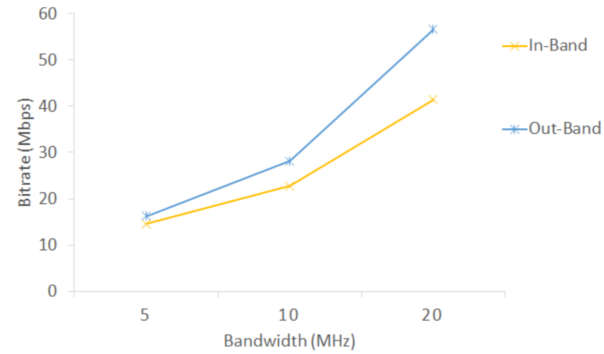


Fig. 13. Downlink throughput for maximum Tx and Rx gains at DeNB and RN.

Figs. 11 and 12 show the throughput results of the self-backhauling outband, i.e., Type 1a RN, network setup for uplink and downlink, respectively. In contrast to the Type 1b solution, where the access and the backhaul link share the same frequency, in the Type 1a solution there is no self-interference, so the throughput of the uplink is better as seen in Fig. 11. Although we obtained a performance enhancement over the conventional LTE network in the downlink when inband solution was used, using outband the performance improvement achieved is even more (see Fig. 12). As we have shown, using a self-backhauling eNB (RN) helps to increase the radio coverage of the conventional LTE network and provides better throughput for cell edge users.

TABLE II
CQI LEVELS FOR OUTBAND SELF-BACKHAULING AT DIFFERENT TX GAINS

Bandwitdh (MHz)	DeNB Gain (dB)					RN Gain (dB)
	70	**75**	**80**	**85**	**90**	**75**
5	9	11	12	14	15	13
10	8	9	11	13	14	12
20	7	8	10	12	14	11

In Table II, the CQI levels for DeNB and RN are shown for Type 1a setup. Note that in the inband solution (Table I) we achieve similar CQI values, which shows the resiliency of the downlink to the self-interference using the inband solution.

Next, we evaluate the scenario where both RN and DeNB use the maximum Tx and Rx gains supported by the SDR platform, i.e., 90 and 125 dB, respectively. Unlike the previous scenario, where the RN cell size was smaller than Donor cell size, in this scenario both cell sizes are targeted to be same. This is representative of the use cases such as quick cell deployment, the need for wireless backhauling due to lack of fiber attachment point, etc.

Fig. 13 shows that achievable downlink throughput goes up to 60 Mbps compared to the 6 Mbps achieved with the direct link in Fig. 8. Although outband solution achieves much higher throughput than the inband one as expected, the inband solution also achieves significant improvement (up to 41Mbps) over the direct link. This shows the benefits of self-backhauling for purposes such as cell densification, or increasing capacity even if the same carrier frequency is used at the donor cell and the relay cell.

VI. Conclusions and Future Work

In this paper, we presented an LTE self-backhauling solution built using an open-source LTE and EPC software and using commodity hardware. Through physical evaluations of this implementation and using an off-the-shelf UE, it was shown that the self-backhauling network allow us to extend the radio coverage of the conventional LTE network. Both outband and inband solutions are evaluated, which show the viability of the developed solution and its performance improvement over the network without the relay node. For future work, the DeNB code can be updated in order to distinguish the incoming packets from a normal UE and a RN, for intelligent resource scheduling solution. In this case, the packets coming from RN optionally would not be encapsulated through GTP to resolve the double GTP encapsulation problem.

Acknowledgment

This work is partially funded by European Union under grant agreement 671551 (H2020 5G-XHaul), 762057 (H2020 5G-PICTURE) and from Spanish Government, MINECO, through project RYC-2013-13029.

References

[1] A. Mourad, "Self-Backhauling in 5G," *5G-PPP Workshop on 5G RAN Design, Air Interface Design and Integration*, Valencia, Spain, Jan. 2016.

[2] C. Hoymann, A. Racz, N. Johansson, and Johan Lundsjo, "A Self backhauling Solution for LTE-Advanced," *WWRF21-WG4-07*, 2008.

[3] Tutorials (OpenAirInterface Usage). [Online]. Available: https://gitlab.eurecom.fr/oai/openairinterface5g/wikis/OpenAirUsage

[4] A. Apostolaras et al., "Evolved user equipment for collaborative wireless backhauling in next generation cellular networks," *IEEE SECON*, 2015.

[5] R. Favraud, N. Nikaein, "Wireless Mesh Backhauling for LTE/LTE-A Networks," *IEEE MILCOM 2015*, pp. 695-700, Oct. 2015.

[6] A. Lo and I. Niemegeers, "Multi-hop Relay Architecture for 3GPP LTE-Advanced," *IEEE MICC*, pp. 123-127, Malaysia, Dec. 2009.

[7] 3GPP. LTE Advanced. Release 10. Available: http://www.3gpp.org/technologies/keywords-acronyms/97-lte-advanced

[8] 3GPP, "TR 36.806 Evolved Universal Terrestrial Radio Access (E-UTRA); Relay architectures for E-UTRA (LTE-Advanced)," 2010.

[9] 5G-PPP XHaul project, https://5g-ppp.eu/5g-xhaul/

[10] A. Anpalagan, M. Bennis, R. Vannithamby, "Design and Deployment of Small Cell Networks," Cambridge University Press, 2015.