Using Opportunistic Caching to Improve the Efficiency of Handover in LTE with a PON Access Network Backhaul

David Stynes, Kenneth N. Brown, Cormac J. Sreenan
CTVR, Dept. of Computer Science,
University College Cork, Ireland

***Abstract*—This paper investigates the converged architecture of an LTE mobile network and a PON access network. We identify that the default handover behaviour for LTE is highly inefficient when performed on a backhaul PON tree-topology. We propose a scheme of intelligent opportunistic caching, using the existing resources at the eNBs, to help mitigate this inefficiency. We show that we can achieve a significant reduction in the amount of redundant traffic sent over the PON and that in circumstances when the PON upstream is heavily congested, we can greatly improve the mobile terminal's sustained data rate by up to 50% during the handover process.**

I. Introduction

Consumers continue to demand ever-increasing amounts of bandwidth, both in wired and wireless situations. This has led to the development of technologies in both fields to provide faster data rates to consumers, but the integration of these two technologies is at times overlooked and combining the two can exhibit some inefficient behaviour.

On the wired side, Passive Optical Networks (PONs) are being deployed to replace existing access networks. A PON is a point-to-multi-point optical access network with no active elements. A variety of different PON technologies have emerged (EPON [1], XGPON [2], LRPON [3], etc..) but they share a common topology. Transmissions are performed between an Optical Line Terminal (OLT) and multiple Optical Network Units (ONUs). Downstream data sent by the OLT is passively broadcast to all ONUs, while upstream data from the ONUs must be scheduled using TDMA [4] and is sent only from each ONU to the OLT. The ONUs cannot directly communicate with each other. PONs provide high speeds (up to 10Gbps for a single wavelength) to a large number of customers (up to about 1000 for a single LRPON).

On the wireless side, the increasing demand has led to the development of broadband wireless services like WIMAX [5] and LTE [6] which support data rates in the range of hundreds of Mbits per second under ideal conditions. To achieve this, it will require the deployment of smaller densely populated cells with high spectral efficiency. In LTE, the base stations at these cells are referred to as e-Node Bs (eNBs). These eNBs must communicate with the Evolved Packet Core (EPC), the core network for LTE, and require high bandwidth, cost-effective backhauling links to the EPC for which PONs have been proposed [7].

During handover in LTE, traffic is forwarded from the source eNB to the target eNB over an X2 link. If using a PON to backhaul traffic, the X2 link would be a logical link between two ONUs in the PON. However, communication between ONUs requires that data must be forwarded up the PON via the OLT to a switch and then back down the PON to the destination ONU. This results in a redundant retransmission down the PON of data that was previously broadcast to all of the ONUs.

In this paper, we propose an opportunistic caching system to reduce the amount of retransmission of data down the PON during LTE handovers. When data is initially sent down the PON intended for a user equipment (UE) currently connected to a source eNB and which is expected to handover in the near future, candidate neighbouring eNBs for the handover, which are connected to the same PON as the source eNB, will cache that data. If the handover occurs, rather than requiring the source eNB to forward data over the logical X2 link, it can then be sent immediately from the cache instead.

II. Related Work

We are not aware of any other proposals to use caching to improve PON-based LTE handovers, so we choose to summarise the most closely related papers.

In [7] the authors identify the peak data rates and latency that must be supported for next generation wireless backhauling links. They also investigate a Digital-over-Fiber converged network where CPRI transmission of LTE signals is performed over an OFDMA-PON with up to 100km fiber spans.

Ranaweera et al. [10] present an algorithm for assigning placement of small cells to utilise an existing Fiber-To-The-Node (FTTN) PON backhaul network. They select the optimal subset of nodes from the FTTN network on which to place small cells to maximise coverage while cost-effectively using the backhaul architecture.

Hussain et al. [11] propose a fully distributed ring-based EPON architecture that enables the support of a converged PON-4G LTE access networking transport infrastructure to backhaul both mobile and wireline multimedia traffic and

978-1-4799-4894-9/14/$31.00 © 2014 IEEE

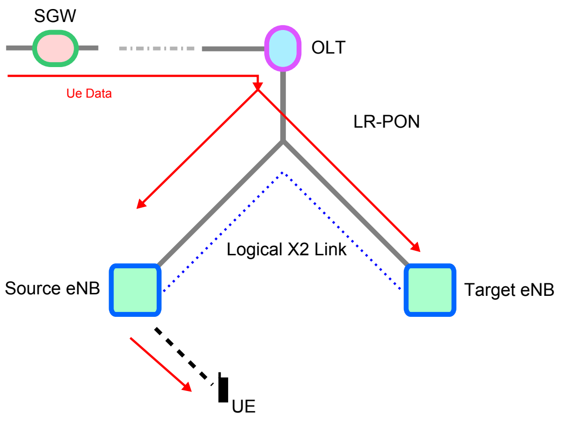

Fig. 1. The state before handover.

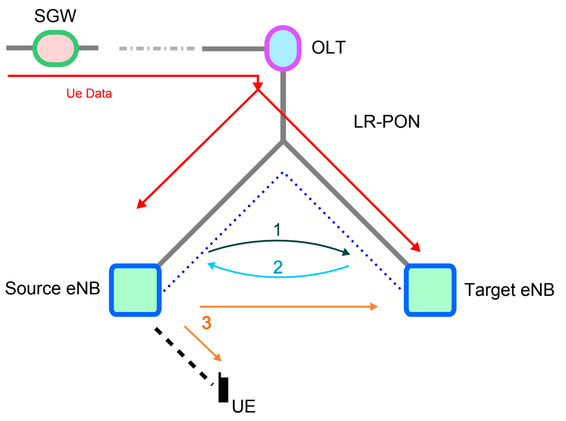

Fig. 2. Initiating Handover.

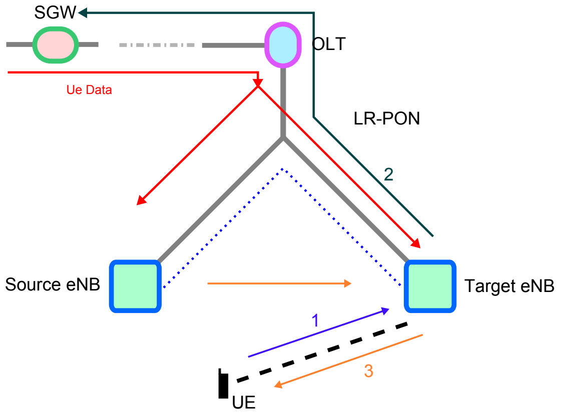

Fig. 3. Finishing Handover.

services. The ring based architecture is proposed to help improve X2 based handovers, but no analysis is given on how the performance of X2 handover will actually change between PONs with a tree topology and a ring topology.

[12] investigate the effect of LTE handover on TCP performance and propose three ways to improve handovers. They show that handover has a significant negative impact on the performance of TCP, increasing the RTT and potentially causing a retransmission timeout expiration of TCP after which the congestion window could drop to 1 segment. They use fast path switch, handover prediction and active queue management to try and maintain the RTT during handover.

III. LTE Handover with a PON Backhaul

We now describe the default behaviour for a LTE handover on PON backhaul without any opportunistic caching. For simplicity we assume that the eNB and the ONUs are integrated into a single entity (typically termed a Cellular Backhaul Unit CBU) from this point forward. If they are implemented as separate entities it does not negatively affect the proposed functionality, although the ONU may need to tag incoming data to indicate to the eNB which ONU, and thus which eNB, the data was originally intended for.

Figure 1 shows the state prior to handover occurring. A UE is connected to the source eNB and is receiving data from the Serving Gateway (SGW), part of the EPC. The logical link between the SGW and the eNB is called the S1 link. Part of the core functionality of a PON infrastructure is that downstream traffic from the OLT to an ONU gets passively broadcast to all ONUs on the PON. Thus, when using a PON as the backhaul link between the SGW and the eNBs, all data sent on the S1 link is actually being passively broadcast to all the eNBs. The source eNB sends the arriving data from the SGW to the UE over the air interface, while all other eNBs would ignore the data as they are not the intended target.

When the source eNB decides a handover is necessary, it contacts the target eNB using the logical X2 link as shown in Figure 2. It 1) requests handover, 2) receives acknowledgement of the request and then if the handover request was accepted, it 3) tells the UE to connect to the target eNB and begins forwarding any buffered data it had intended for the UE to the target eNB over the logical X2 link. The logical X2 link is also implemented across the PON backhaul connection. Since the PON provides no direct link between ONUs, this forwarding requires sending data up the PON to a switch via the OLT and then back down the PON again to the target eNB. The source eNB no longer sends any data to the UE after telling it to handover to the target eNB, data received by the source eNB from the SGW on the S1 link during the handover process is also forwarded over the X2 link.

As shown in Figure 3, the target eNB then 1) accepts the connection from the UE, 2) sends the EPC a *Context Release* message, informing it that the handover has occurred so that the SGW no longer sends data intended for the UE to the source eNB and 3) begins forwarding data to the UE, starting with any data that it has to wait to receive from the source eNB. After receiving the context release message, the EPC sends an *End Marker* to the source eNB, letting it know the SGW has finished sending it data and that it can now free up the resources that were reserved for the UE.

The handover process is now completed, but source eNB must *continue forwarding* its remaining data packets to the target eNB. If this forwarding process is slow (e.g. the upstream/downstream on the PON is congested) then the UE can experience greatly increased delays for packet arrival as a result of the handover.

IV. Caching during Handover

We propose to perform opportunistic caching at the target eNB to reduce the amount of data that must be forwarded over the X2 link. To do this, it is necessary that the target eNB be able to read data intended for the source eNB. Typically, a PON will encode data specifically for each ONU and so we assume that all the eNBs for the same mobile provider will use a shared encryption key on the PON. The source eNB must detect when a handover is growing increasingly likely, and send a warning message over the X2 link to the possible target eNBs that they should begin caching. Our focus in this paper is to define the caching mechanism and evaluate its potential, leaving issues such as selection of target eNBs and caching policy for future work.

Once the target eNB begins caching, it will store as much data as possible which is intended for the source eNB's UE. Then, once the source eNB initiates the handover, the target eNB can inform it of which data it has cached and the source eNB need only forward any data which is not contained in the cache. The caching uses a shared memory space with the data being buffered for the active current UEs of the eNB. If more memory space is necessary for the active UEs, it is possible to free up the cached memory and request it to be forwarded from the source eNB instead. Thus, it simply reverts to the default handover without caching if no memory is available to cache and is never worse than the default non-caching approach.

Data sent on the S1 link by the SGW during the handover can also be buffered at the target eNB and will not need to be sent by the source eNB either. In the event of the target eNB running out of storage space for this S1 data, it can request the data from the source eNB instead. We assume that the target eNB will always have enough space to buffer this S1 traffic once the handover has begun, since in default LTE the source eNB forwards it all with that assumption.

A. Cache Management

We explore three different methods of managing the limited cache space available at the target eNB for a single UE, to compare against the performance of standard LTE.

INFINITE - Caching when we assume that every single byte is already cached at the target eNB. Used as an optimal caching to compare against the other methods.

FIFO - The target eNB cache is operated as a FIFO queue, removing the first element whenever the cache is full and a new packet arrives.

START - Caching where we attempt to store data in the cache starting from the first packet currently queued to be transmitted to the UE by the source eNB when the handover occurs. When this cache is full, newly arrived packets are discarded. There is periodic communication from the source eNB to the target eNB to tell it which packets have already been sent to the UE. When this message arrives, any redundant packets that were already sent to the UE are removed from the cache.

NONE - The default behaviour of LTE, in which there is no caching performed.

Note that the START scheme can result in gaps in the sequences of data packets in the cache due to its nature of discarding newly arrived packets when the cache is full.

All of these cache management schemes have an additional overhead of control message passing required between the source and target eNBs on the X2 link. It is possible to include some of the messages as part of the existing control messages of the X2-Handover process, at a modest increase to their size. The additional messages for the different schemes are as follows:

a) 'Last Packet Sent To UE' Message: After receiving the Handover Acknowledgement, the source eNB must send a message informing the target eNB which was the last packet it sent to the UE. This must be sent for all types of caching. It is the only additional overhead for INFINITE.

b) 'Packet Request' Message: The target eNB must send this message to request the packets missing from the cache that need to be forwarded. For FIFO, it can send a message identifying the first packet it has so the source eNB knows to forward all packets sequentially before that one. For START, a larger message must be sent containing the start and end points of the subsequences of packets it is missing.

c) 'Final Packet Forwarded On X2' Message: The source eNB must send this message so that the target eNB can identify when to start sending from its cache. The START method may require multiples of this message to be sent, if many fragments of the sequence were requested. FIFO only requires one of these messages.

d) 'Cache Update' Message: For START only, the source eNB must periodically send this message to inform the target eNB of which packet was last sent to the UE. The more frequently it is sent the more likely the contents of the cache will be usable, but also the greater the overhead, though this overhead is always low relative to the size of the user data.

V. Benefits of Caching

There are two benefits to caching. Firstly, the amount of data required to be sent up and back down the PON is reduced by the size of the non-redundant data stored in the cache. Secondly, in circumstances where the upstream on the PON is congested, there can be an improvement in the data rate experienced by the UE.

A. Calculating the Time for Handover to Complete.

In this section we analyze the expected time for the UE to receive all of the data that was buffered at the source eNB and also the data sent by the SGW on the S1 link during the handover. We will use this analysis in Section VI for evaluation purposes.

For the given UE which is performing the handover, we define the following constants:

Tx_{Down} - The maximum data rate available to the UE on the downlink of the PON.

Tx_{Up} - The maximum data rate available to the UE on the uplink of the PON.

Tx_{Air} - The maximum data rate available to the UE on the radio interface.

Tx_{Sgw} - The rate the SGW is sending data to the UE, it is assumed that $Tx_{Sgw} \geq Tx_{Air}$.

All rates are measured in Bs^{-1} (Bytes per second).

$Buff_{Source}$ - The amount of unsent bytes waiting in the buffer at the source eNB when the handover ACK was received.

$Cache_{FIFO}$ - The amount of non-redundant cached bytes waiting in the cache at the target eNB when the handover ACK was received, when using FIFO.

$Cache_{START}$ - The amount of non-redundant cached bytes waiting in the cache at the target eNB when the handover ACK was received, when using START.

B_{S1} - The bytes arriving at the source eNB on the S1 link during the handover. Equal to Tx_{Sgw} times the period of time between source eNB receiving the handover ACK and the SGW receiving the *Context Release* message.

Non-redundant cached bytes are those which have not yet been sent to the UE by the source eNB. We assume that Tx_{Down} is fast enough that no packets get delayed/dropped in the downlink direction, since otherwise the behaviour is unpredictable when control messages may be delayed or lost. i.e.

$$Tx_{Down} > (Tx_{Sgw} + Tx_{Up})$$

The total time taken for the UE to receive the bytes from the source eNB is composed of two parts:

T_{Buff} - The time taken for the UE to receive the $Buff_{Source}$ bytes that were waiting in the buffer at the source eNB when the handover was confirmed.

T_{S1} - The time taken to receive the bytes that arrived at the source eNB after the handover was confirmed, until the context release was performed.

1) Calculating T_{Buff}: The time for the packets in the source eNB buffer, at the start of the handover, to reach the UE for NONE:

$$T_{Buff} = \frac{Buff_{Source}}{Min[Tx_{Up}, Tx_{Air}]}$$

for INFINITE:

$$T_{Buff} = \frac{Buff_{Source}}{Tx_{Air}}$$

for FIFO:

$$T_{Buff} = \frac{(Buff_{Source} - Cache_{FIFO})}{Min[Tx_{Up}, Tx_{Air}]} + \frac{Cache_{FIFO}}{Tx_{Air}}$$

For START, we first determine the amount of packets that can be sent over the X2 link while the cached packets at the target eNB are being sent to the UE over the radio interface. When $Tx_{Up} < Tx_{Air}$, this can be represented with the following infinite series. The first term is the amount of bytes that can arrive over the X2 link before the cache at the target eNB is all sent to the UE. Every other term is the number of additional bytes that can arrive over the X2 link while the bytes of the previous term are being sent to the UE over the radio interface.

$$s = Cache_{START}(\frac{Tx_{Up}}{Tx_{Air}}) + Cache_{START}(\frac{Tx_{Up}^2}{Tx_{Air}^2}) + Cache_{START}(\frac{Tx_{Up}^3}{Tx_{Air}^3}) + ...$$

Solving the infinite series, we get:

$$s = Cache_{START}(\frac{Tx_{Up}}{Tx_{Air} - Tx_{Up}}), if Tx_{Air} > Tx_{Up}$$

The total time for START then depends on whether the amount of bytes in $Buff_{Source}$ is less than, equal to or greater than the value of s.

If $Tx_{Up} \geq Tx_{Air}$ or $(Buff_{Source} - Cache_{START}) < s$ then:

$$T_{Buff} = \frac{Buff_{Source}}{Tx_{Air}}$$

Else:

$$T_{Buff} = (\frac{Cache_{START}}{Tx_{Air}}) + (\frac{((Buff_{Source} - Cache_{START}) - s)}{Min[Tx_{Up}, Tx_{Air}]}) + (\frac{s}{Tx_{Air}})$$

2) Calculating T_{S1}: The time taken for the UE to receive the B_{S1} bytes is for NONE:

$$T_{S1} = \frac{B_{S1}}{Min[Tx_{Up}, Tx_{Air}]}$$

and for all types of caching:

$$T_{S1} = \frac{B_{S1}}{Tx_{Air}}$$

since when caching we assume that the target eNB reserves enough space to buffer the bytes arriving on the S1 link.

3) Calculating the Improvement for the UE: The improvement, I, in bytes received by the UE can then be measured as:

$$I = (T_{NONE} - T_{CACHE}) \times Tx_{Air}$$

Where T_{NONE} is the total time to transfer all data from the source eNB when not caching, and T_{CACHE} is the time taken for a chosen caching scheme.

VI. Experimental Evaluation

In order to validate our basic premise that opportunistic caching can be beneficial and is feasible, we conducted a simulation of a simple topology using the NS-3 simulator [8] and LENA module for LTE [9]. We have implemented an approximation of a PON backhaul used to communicate between the eNBs and the SGW, with a structure as shown in Figure 4. The changes we have made to the default LTE module are as follows: we modify the point-to-point S1 links between the SGW and each eNB, changing them to only send traffic in the downstream direction. We add a new point-to-point link between each eNB and the SGW for the upstream traffic. We remove the direct X2 link between eNBs and instead force them to route packets through the SGW to send data to the other eNB. Any time the SGW receives a packet intended for either of the eNBs (from either an eNB or the remote host), it duplicates that packet and sends a copy of the packet down both of the downlinks to both eNBs at the same time. This emulates the passive broadcasting ability of a PON. In the upstream direction in our simulation, it is possible for both eNBs to transmit simultaneously, which is not possible on a PON. However, we minimise the amount of uplink traffic generated by the target eNB, by only transmitting the small control packets that are absolutely necessary.

In our experiments, the remote host immediately starts to send data intended for the UE, which is forwarded through the SGW and the source eNB that the UE is initially connected to. We use a simple UDP traffic model to allow us to focus on the caching behaviour. The remote hosts sends bursts of five 1024 byte packets to the UE at a fixed interval i. The UE is stationary at all times to ensure fixed data rates between it and each eNB. After 0.3seconds, the source eNB initiates the handover procedure for the UE. We have the caching active at the target eNB at all times, though for practical application it should only be triggered once a handover is expected to occur soon. The total duration is 1 second of simulated time.

We have varied the bandwidth of the uplink point-to-point links to emulate the case where there is heavy upstream traffic and the eNB cannot transfer at the maximum speed desired. Figure 5 shows the effect of varying the cache size when the uplink speed is capped at 1Mb/s, and the remote host is sending a burst of five 1024 bytes packets every 10 ms. These simulation results for packets arrived at the UE were within a 3% margin of error from the expected number by our formulae for FIFO and START. At this speed, the uplink is slower than the speed of the target eNB's radio interface to the UE and so caching can provide a significant boost to the delivery time of the packets. We see that the START cache management scheme consistently outperforms FIFO at all cache sizes, since it starts sending immediately while FIFO must wait for some packets to be forwarded by the source eNB first. Both schemes provide a significant improvement, of up to 50% more packets received by the UE in the same time frame, over not caching.

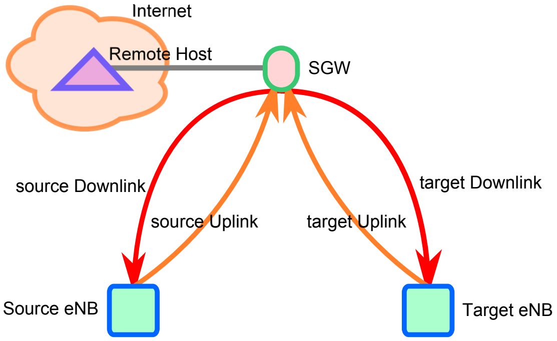

Fig. 4. Topology of the Simulation Model.

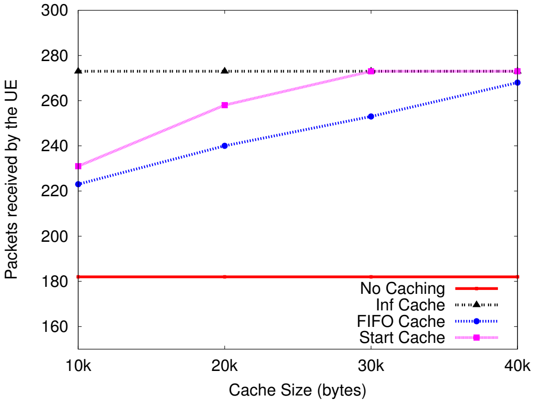

Fig. 5. Packets received by the UE in a congested uplink. Where i = 10ms, $uplinkMaxRate = 1Mb/s, downlinkMaxRate = 100Mb/s$.

The INFINITE scheme shows the optimal performance boost we can expect to achieve when every necessary packet is already cached at the target eNB when handover is initiated. At larger cache sizes START and FIFO can reach this optimum, with START reaching it far sooner since it can begin sending data to the UE immediately after it connects to the target eNB and receive forwarded data while sending the cached data. Thus, START does not require to have every packet stored in the cache in order to be able to perform as well as INFINITE does in the metric of received packets at the UE.

We do not show the results for the case where the downlink from the SGW to the eNBs is congested, as delays and dropping of control messages, both for not caching and caching, rendered the test results difficult to compare. Caching performs better as long as the additional control messages do not get dropped, therefore START with its additional messaging requirements is not recommended if packet loss is high.

Figures 6 and 7 depict when the upstream and downstream PON are not congested, showing the amount of redundant

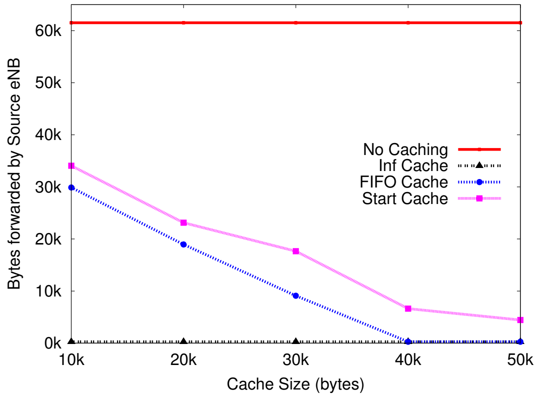

Fig. 6. Bytes forwarded by the source eNB to the target eNB for various cache sizes. Where $i = 10\text{ms}, uplinkMaxRate = 100Mb/s, downlinkMaxRate = 100Mb/s$.

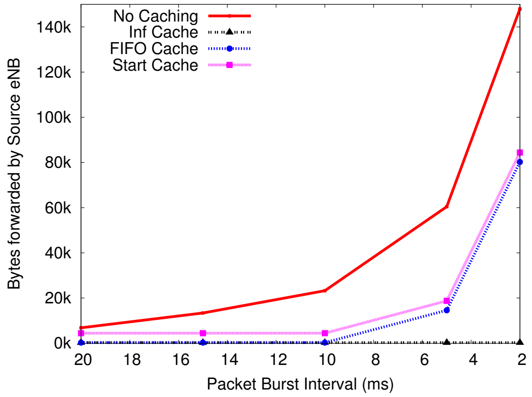

Fig. 7. Bytes forwarded by the source eNB to the target eNB for various data rates from the remote host. Where $cacheSize = 25k, uplinkMaxRate = 100Mb/s, downlinkMaxRate = 100Mb/s$

bytes getting sent down the PON as part of the forwarding process for varying cache sizes and varying intervals between packet bursts respectively. Our additional control messages are also included as part of the Bytes Forwarded metric. The up and down links were both set to speed 100 Mb/s. In Figure 6 the remote host sent a burst of packets to the UE every 10ms, while for Figure 7 the cache size was fixed at 25k bytes. Both FIFO and START again consistently reduce the amount of traffic on the PON, and at high enough cache sizes FIFO is equivalent to INFINITE. START can never perform as well as INFINITE or FIFO however due to the additional 'Cache Update' control messages, and since it isn't possible to ever keep it so up to date that every packet in the cache is useful.

VII. Conclusions

We have shown that there is a large amount of redundant data transmission as part of the LTE handover process when using a PON backhaul. We have demonstrated the value of using caching to reduce this amount of data transfer. The caching can be performed using the existing memory used by eNBs for buffering data to their connected UEs, with priority given to the needs of the current UEs. When it is not possible to cache, the behaviour can revert to the same as default LTE handover without caching, so attempting to cache will never perform worse than not caching. In cases where the uplink is congested, we have shown a significant improvement of up to 50% in the number of packets delivered to users during handover. This can be expected to have an even greater effect since packet delays can heavily impact performance for TCP, which is quick to reduce the speed packets are sent when delays are detected.

For uplink congested situations, START is the best choice for improving the quality of experience for the end users. FIFO is the best for maximising the efficiency of the PON by reducing the amount of data transmitted on it. It should be possible for the eNBs to monitor the status of the PON and decide which type of caching to use based upon its activity.

In future work, we will investigate more advanced cache management algorithms for when multiple UEs are performing handover and sharing the same memory space at the target eNB, and we will focus on scalability for larger networks. We will explore the effect of different kinds of traffic, such as TCP and QoS-sensitive traffic types. Finally, we are developing techniques for intelligent triggering of caching at specific eNBs, based on observed mobility patterns and experiments with real traffic traces.

References

[1] G. Kramer, B. Mukherjee, and A. Maislos, *Ethernet Passive Optical Network (EPON): a missing link in an end-to-end optical internet*, Multi-Protocol Over WDM: Building the Next Generation Internet, S. Dixit, Ed. New York: Wiley, Mar. 2003.

[2] F. J. Effenberger, *The XG-PON System: Cost Effective 10 Gb/s Access*, J. Lightwave Technol., Vol. 29, No. 4., pp. 403-409, Feb. 2011

[3] H. Song, K. Byoung-Whi and B. Mukherjee, *Long-reach optical access networks: A survey of research challenges, demonstrations, and bandwidth assignment mechanisms*, IEEE Communications Surveys and Tutorials, Vol.12, No.1, pp. 112-123, First Quarter 2010

[4] G. Kramer, B. Mukherjee and G. Pesavento, *Interleaved polling with adaptive cycle time (IPACT): a dynamic bandwidth distribution scheme in an optical access network*, Photonic Network Communications, Vol. 4, No. 1, pp. 89-107, 2002

[5] S. Ahmadi, *An overview of next-generation mobile WiMAX technology*, IEEE Communications Magazine, vol. 47, pp. 84-98, 2009

[6] D. Astely, E. Dahlman, A. Furuskar, Y. Jading, M. Lindstrom, and S. Parkvall, *LTE: the evolution of mobile broadband - LTE part II: 3GPP release 8*, IEEE Communications Magazine, vol. 47, pp. 44-51, 2009

[7] M. Milosavljevic, P. Kourtessis, Wansu Lim and J.M. Senior, *Next generation PONs with wireless backhauling*, 13th International Conference on Transparent Optical Networks (ICTON) Vol. 1, No. 4, pp.26-30, June 2011 doi: 10.1109/ICTON.2011.5970782

[8] *NS-3 The Network Simulator*, http://www.nsnam.org/

[9] *The LTE-EPC Network Simulator (LENA) project*, http://iptechwiki.cttc.es/LTE-EPC_Network_Simulator_(LENA)

[10] C. Ranaweera, M.G.C. Resende, K.C. Reichmann, P.P. Iannone, P.S. Henry, B-J. Kim, P.D. Magill, K.N. Oikonomou, R.K. Sinha, and S.L. Woodward, *Design and optimization of fiber-optic small-cell backhaul based on an existing fiber-to-the-node residential access network*, IEEE Communications Magazine, pending, 2013

[11] S. Hussain, S. R. Zaidi, and M. A. Ali, *Next Generation PON Based LTE Architecture and Challenges in Femtocells Deployments* 2012 International Journal of Research in Wireless Systems (IJRWS), Vol. 1, No. 2, 2012

[12] D. Pacifico, M. Pacifico, C. Fischione, H. Hjalrmasson, and Johansson, K.H., *Improving TCP Performance During the Intra LTE Handover*, IEEE Global Telecommunications Conference GLOBECOM, pp. 1-8, 2009