Evaluating Dynamic Bandwidth Allocation of Virtualised Passive Optical Networks Over Mobile Traffic Traces

Pedro Alvarez, *Student Member, IEEE,* Nicola Marchetti, *Senior Member, IEEE,* and Marco Ruffini, *Senior Member, IEEE*

***Abstract*—Deploying Passive Optical Networks (PONs) requires substantial investments, despite their savings compared to point-to-point solutions. Consequently, network-sharing business models, where a network operator leases its system to other service providers, are being investigated and deployed in some countries. These types of schemes require that operators are either physically isolated or have reserved capacity, so that they can operate independently. We present the benefits that our group-assured Dynamic Bandwidth Assignment (DBA) algorithm brings to access network virtualisation. This algorithm enables operators to take advantage of the benefits of statistical multiplexing, while maintaining isolation from other operators in the PON. To test our algorithm, we considered the particular use case of a mobile system backhauling multiple basestations through a PON. Our tests, operated over real mobile traffic traces, show increased Quality of Service (QoS) with a reduction of backhaul capacity of over 30% compared to a legacy DBA algorithm.**

***Index Terms*—XG-PON, DBA, group-assured bandwidth, PON virtualisation.**

I. Introduction

IN telecommunications networks, the access links connecting the operator's equipment to the operator's customers – also known as the last mile – is often seen as the performance bottleneck, due to legacy copper infrastructures which limit customers' bandwidth and reachability. Bringing fibre closer to the customer, in a Fibre To The *x* (FTTx) architecture, is a possible solution, but it requires heavy investments to replace the copper infrastructure by optical fibre.

This creates a problem of financial viability for fibre access networks, which has delayed their deployment for decades [1], [2]. Although Passive Optical Networks (PONs) reduce the cost of deploying FTTx, by sharing electronic equipment and optical fibre among many customers, PON deployment still requires a considerable investment. For this reason, it is important that access networks can be shared across two dimensions: the service dimension, allowing multiple services to coexist on the same network – such as residential broadband services, small and medium business services, as other services like mobile backhaul – and the ownership model dimension, allowing multiple service providers to operate over the same physical infrastructure.

Pedro Alvarez, Nicola Marchetti and Marco Ruffini are with CONNECT, Trinity College Dublin, the University of Dublin, Dublin 2, Ireland. e-mail: pinheirp@tcd.ie.

Manuscript received October 15, 2015

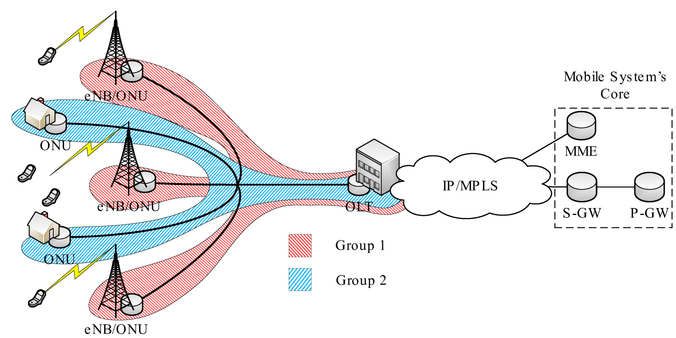


Fig. 1. Grouped connections in a PON with a tree architecture.

Sharing an access network across different services and providers requires virtualising the network to allow for independent operation and fair access to network capacity [3]. Virtualisation enables allocating network slices to different service providers, and within a service provider to different services. Our work tackles the problem of PON virtualisation by proposing an algorithm that runs at the Optical Line Terminal (OLT), i.e., the PON network head end, enabling partitioning of the PON upstream capacity among virtual slices. The idea consists in grouping the assured capacity normally assigned to a number of users individually into a group-assured capacity assigned to a virtual slice. This ensures that all unused capacity within a virtual slice is first re-assigned among users of the same slice. The capacity that is not required by the users in the slice is then redistributed among other slices in the same PON. This is visually illustrated in Fig. 1 where two different service providers, a mobile operator and a broadband provider, are assigned a slice of the PON's assured capacity.

In this article, we illustrate the benefits that can be achieved with group-assured bandwidth, as implemented through our proposed algorithm. To do this, we evaluate the algorithm through the simulation of a 10-Gigabit-capable Passive Optical Network (XG-PON) under different scenarios.

Initially, we study the network when artificial, homogeneous load was offered to the customers. This shows that group-assured bandwidth can improve the delay and packet loss for the same amount of assured capacity, even when the customers' average throughput is the same. This also shows a trend: the more capacity is shared, the bigger the improvements obtained. We note that, even though improvements are possible in homogeneous scenarios, real traffic will be heterogeneous, making the benefits more substantial.

We then focus on the use case of a mobile operator backhauling a number of micro cells through a shared XG-PON, showing the benefits that our group-assured algorithm can bring to operators, under realistic scenarios and realistic traffic conditions, i.e. using real mobile data traffic from an Irish operator. Since the algorithm's performance is strongly dependent on the disparities between the base stations' loads [4], we consider the use of real traffic data of paramount importance.

Our results show that our proposed algorithm can bring substantial benefits to heterogeneous patterns (i.e., presence within the virtual slice of both highly and lightly loaded users) typical of mobile traffic, and can reduce the requirement on peak backhaul capacity by over 30%.

II. The Group-Assured bandwidth concept

In the typical tree architecture of PONs, illustrated in Fig. 1, a single fibre is shared among many customers through the use of passive optical splitters. Upstream traffic is scheduled by the OLT through a Dynamic Bandwidth Assignment (DBA) algorithm, which allocates capacity to the Optical Network Units (ONUs), i.e. the PON tree leaf nodes. To differentiate services, logical connections called XGEM-Ports are established between the OLT and the ONUs in the upstream direction. XGEM-Ports belonging to the same ONU will be grouped into logical containers named Transmission Containers (T-CONTs), the logical unit in XG-PON to which the capacity gets allocated. If multiple XGEM-Ports exist in a T-CONT, then the ONU must decide how to divide the assigned capacity amongst them.

In XG-PON, Quality of Service (QoS) is provided by defining four types of bandwidth: fixed, assured, non-assured and best effort. Of these four, fixed and assured bandwidth are guaranteed to be assigned, but they differ in that fixed is always assigned regardless of being needed, while assured is only assigned if there is traffic that needs transmission. The other two types – non-assured and best effort – are only assigned when spare capacity is available, thus giving no guarantees of QoS. The difference between the two types is in the order in which they are assigned, where non-assured has higher priority.

In PON protocols, after the assured capacity has been assigned, the remaining capacity is typically re-scheduled among all T-CONTs. Our proposed algorithm tackles this situation that is not optimal in a virtualised access architecture: since a mobile operator is charged for the amount of assured capacity purchased for backhauling its base stations, it is desirable that any spare capacity arising from its group of base stations is first re-distributed among its own group. If after this assignment there is still unused capacity, this is then reassigned among the entire PON. In this paper we use an extension of the GIANT algorithm [5], which we call Group GIANT (gGIANT) originally introduced in [6].

In our algorithm, T-CONTs are grouped into group Transmission Containers (gT-CONTs), a logical set that identifies which T-CONTs belong to which groups. The T-CONTs considered are then assigned individual assured bandwidths, which dictates the minimum amount of capacity that they are assured. To allocate group bandwidth in this algorithm, if any of the T-CONTs does not need some of the bytes reserved for it in a frame, those bytes can then be allocated to another T-CONT that belongs to the same gT-CONT. If no T-CONT within the gT-CONT's group needs it, then these bytes are available for non-assured and best effort bandwidth, just like normal GIANT. In Fig. 2, it is illustrated the upstream logical connections of XG-PON, when gT-CONTs are considered.

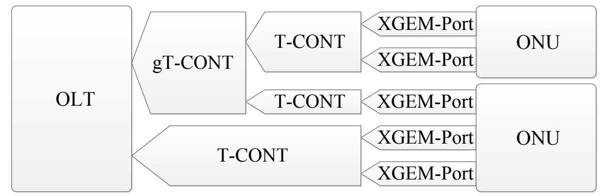


Fig. 2. Upstream Logical Connections, including gT-CONTs.

To the author's best knowledge there are no previous works in grouping T-CONTs with the objective of sharing assured bandwidth across members of a common entity. There has been, however, some work in grouping T-CONTs with other objectives. For example, in [7] a DBA mechanism was proposed where T-CONTs are grouped into two groups, either by considering their priorities or by dividing the T-CONTs in half. This work has the objective of reducing the report idle time, i.e. the time where the OLT waits for all buffer reports to arrive to make a decision, to reduce the problem of queue state inconsistency. Other authors have addressed the issue of sharing assured bandwidth across the whole PON. In [8], [9], DBA mechanisms were proposed that share assured bandwidth among all T-CONTs, which makes more efficient use of the assured bandwidth. However, this does not consider the case where an operator may want to assure bandwidth to a group of ONUs. With group-assured bandwidth, a customer with multiple ONUs in the same PON (e.g. a mobile operator) can make use of the benefits of statistical multiplexing, while maintaining isolation from the other users of the PON. This isolation is an important feature, since these users are possibly from competing operators.

III. Group GIANT Dynamic Bandwidth Assignment Algorithm

In GIANT, each bandwidth type is characterized by two parameters: the service interval and the allocation bytes. The service interval, specified in frames, dictates how often the T-CONT gets served, while the allocation bytes dictate how many bytes on the upstream frame can be assigned to the T-CONT. To know when to assign an upstream transmission, the DBA engine will also keep a counter per bandwidth type, that is decreased every upstream frame. When this counter expires, the OLT grants a transmission to the T-CONT and the counter is reset to its service interval value.

In gGIANT, the scheduler needs to be capable of assigning group-assured bandwidth by sharing unused capacity from individual T-CONTs of the same group. To do this, the DBA engine keeps a list of all the T-CONTs in a gT-CONT. Each gT-CONT will keep a byte counter, which keeps track of the amount of bytes unused from the individual T-CONTs at each upstream frame.

To perform the DBA process, first fixed and individual-assured bandwidth are assigned as per the GIANT algorithm. The only difference is, when assigning individual-assured bandwidth, if a particular T-CONT did not need all of the bytes reserved for it in the upstream frame, the amount that was not needed is added to the group's byte counter, to know how many bytes are available to the group.

After assigning individual-assured bandwidth, the DBA engine will go through all the groups, checking how many bytes are available from the previous stage. Unused bytes are then assigned to a T-CONT from the group that needs them, in a round robin fashion.

After finishing the assignment of group-assured bandwidth, non-assured and best-effort bandwidth follow, similarly to the GIANT scheduler. Note, that if there is leftover capacity after the assignment of group-assured bandwidth, then the unused bytes become eligible for non-assured and best-effort assignment, as usual.

This process is illustrated in Fig. 3, where four T-CONTs belonging to the same gT-CONT are depicted. We can see on the first frame, that T-CONT 1 does not use all the bytes reserved for it and so some are assigned to T-CONT 3. Two frames later, it is T-CONT 2 which does not use all the bytes and so some are assigned to the next T-CONT in the group, T-CONT 4.

We show the pseudo-code for gGIANT algorithm in Procedures 1, 2 and 3. Here, Procedure 1 is the main procedure, which is called every 125 μsec to generate the *BwMap* message, which is used to tell all the ONUs when to transmit. In this procedure, bandwidth is assigned in a loop until the upstream frame is full or all bandwidth types (fixed, assured, etc.) of all T-CONTs have been served. This loop will call the $GetNextTcontParameter()$ function to obtain the next T-CONT parameter; a structure that contains the T-CONT to be served, the type of bandwidth being served, the allocation bytes and the service interval of the particular connection and type of bandwidth.

The $GetNextTcontParameter()$ will get the bandwidth parameters in the following order of the priorities: Fixed, Individual Assured, Group Assured, Non-Assured and Best Effort. When $GetNextTcontParameter()$ reaches the group assured parameters, it will go through the list of T-CONT in the group in a round robin manner. Only when all the individual T-CONTs in a group have been served, $GetNextTcontParameter()$ will move on to the next group parameter.

Depending on the type of bandwidth obtained from the $GetNextTcontParameter()$ different functions will be called to allocate the bandwidth. We will focus on describing $allocateIndivAssuredBandwidth()$ and $allocateGroupAssuredBandwidth()$ since they differ from the GIANT algorithm.

Procedure 2 is the procedure used to assign individual-assured bandwidth. In this procedure, first the allocation bytes, the current counter value, and the T-CONT queue length are obtained. Then, when the timer has expired, the allocation bytes are compared with the requested queue size: in case the number of bytes in the queue exceeds the allocation bytes, all the capacity of the T-CONT is used; in the case it does not exceed it, and the T-CONT belongs to a gT-CONT, the unused bytes are added to the shared byte counter of the gT-CONT.

Procedure 1 gGIANT algorithm

```
while Upstream frame is not full do
    tcontParam = GetNextTcontParameter()
    bwType = GetBandwidthType(tcontParam)
    if bwType == FIXED_BW then
        allocateFixedBandwidth(tcontParam)
    else if bwType == INDIV_ASSURED_BW then
        allocateIndivAssuredBandwidth(tcontParam)
    else if bwType == GROUP_ASSURED_BW then
        allocateGroupAssuredBandwidth(tcontParam)
    else if bwType == NON_ASSURED_BW then
        allocateNonAssuredBandwidth(tcontParam)
    else if bwType == BEST_EFFORT_BW then
        allocateBestEffortBandwidth(tcontParam)
    end if
    if all bandwidth types of all T-CONTs were served then
        break
    end if
end while
UpdateAllTimers()
ClearSharedBytesCounters()
UpdateGroupRoundRobinPointers()
FinalizeBwMapProduction()
```

Procedure 2 AllocateIndividAssuredBandwidth

```
tcont = GetTcont(tcontParam)
gtcont = GetGroupTcont(tcontParam)
allocBytes = GetAllocationBytes(tcontParam)
timerValue = GetCounterValue(tcontParam)
buffOcc = GetBufferOccupation(tcont)
if timerValue == TIMER_EXPIRE then
    if buffOcc! = 0 then
        if buffOcc > allocBytes then
            sizeToAssign = allocBytes
        else
            sizeToAssign = buffOcc
            if gtcont! = 0 then
                unusedBytes = allocBytes − buffOcc
                AddSharedBytes(gtcont, unusedBytes)
            end if
        end if
        Assign(tcont, sizeToAssign)
    end if
end if
// If the timer has not expired or there are no bytes in the
// buffer no bytes are granted to this T-CONT in this frame
```

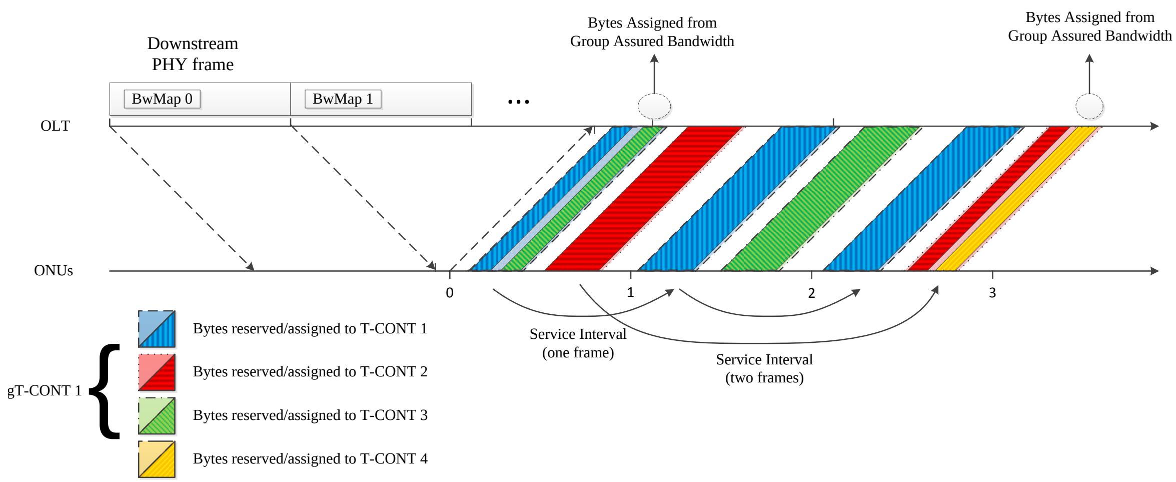


Fig. 3. gGIANT DBA Algorithm

Finally, in Procedure 3 group-assured bandwidth is assigned. This procedure will get the amount of shared bytes from the gT-CONT and assign them if the selected T-CONT has bytes on the queue. It should be noted that if the ONU of the selected T-CONT has not been served before, i.e., if a new burst is necessary, PHY overhead such as inter-gap spacing and preamble must be taken into account when assigning the shared bandwidth. When calculating the amount of bytes allocated to this T-CONT, this overhead must be deducted from the available bytes, and when calculating how much was used, the overhead must be taken into account.

Procedure 3 AllocateGroupAssuredBandwidth

$gtcont = GetGroupTcont(tcontParam)$
$tcont = GetCurrentTcont(gtcont)$
$sharedBytes = GetSharedBytes(gtcont)$
$buffOcc = GetBufferOccupation(tcont)$
$overhead = phyOverhead + XgtcHeader + XgtcTrailer$
if $sharedBytes > 0$ && $buffOcc > 0$ **then**
  if $CheckServedTcont(tcont)$ **then**
    $sizeToAssign = min(buffOcc, sharedBytes)$
    $DecreaseSharedBw(sizeToAssign)$
  else
    $availableBytesToShare -= overhead$
    $sizeToAssign = min(buffOcc, sharedBytes)$
    $DecreaseSharedBw(sizeToAssign + overhead)$
  end if
  $Assign(tcont, sizeToAssign)$
end if

Regarding the computational complexity, it is important to note the differences between GIANT and gGIANT. The GIANT algorithm was successfully implemented in [5] and gGIANT is based on it with some additions. Specifically, minor changes were made in Procedure 2 and a new procedure, Procedure 3, was added. In Procedure 2, it was required an extra-comparison to check if the T-CONT belongs to a group, a subtraction to calculate how many spare bytes exist after the allocation, and an extra addition to add the non assigned bytes to the group byte counter. In Procedure 3, four extra comparisons were needed: one to check if the T-CONT has data in the buffer, one to check if there are any shared bytes available, one to check if the T-CONT was served before, and one to check the minimum between the buffer occupancy and the shared bytes counter. Besides this, two extra subtractions were required: one to discount overhead bytes from the shared bytes counter, another to decrease the shared bytes counter after an assignment. Finally, one addition is done to add the granted bytes to the *BwMap*.

In the worst case scenario, all T-CONTs in the PON will belong to a group and these extra operations will be called for every T-CONT. This means an increase in simple operations, such as additions, subtractions and assignments, that in the worst case could be between two and three times those of the GIANT algorithm. However, the complexity of gGIANT will still grow linearly with the number of T-CONTs, as in the original algorithm, which means that gGIANT's scalability is still assured.

IV. Performance Evaluation of gGIANT under Artificial Loads

To evaluate the performance of the gGIANT scheme, we used the XG-PON module for the ns-3 simulator, a C++, open-source, event-driven, network simulator [10]. This module was developed at our research centre with the objective of providing a free and open-source research platform to study XG-PON. A detailed description of this module is available in [11] and the code can be downloaded from the project's website[1].

First, we analyzed the average delay and packet losses when the load of all the ONUs was increased equally. In these simulations, the network had one gT-CONT with n T-CONTs and 64 ONUs, each ONU with one T-CONT. Each

[1] http://sourceforge.net/projects/xgpon4ns3/

TABLE I
XG-PON SIMULATION PARAMETERS

Number of ONUs	64
T-CONTs per ONU	1
XGEM-Ports per T-CONT	1
T-CONTs' Service Interval	4 frames
Fixed Bandwidth per T-CONT	64 kbps
Assured Bandwidth per T-CONT	38.08 Mbps
Fibre transmission delay	0.4 msec
Use of FEC in upstream T-CONTs	No FEC
Queue Size	100 KiB
Simulated Time	50 sec
Application Traffic Type	Poisson
Transport Protocol	UDP
Packet Size (Bytes)	64 (60%), 500 (20%), 1500 (20%)

individual T-CONT had both fixed and individual-assured bandwidth: the fixed bandwidth was used only to transmit upstream buffer status reports periodically, while the assured bandwidth took care of transporting the application data. Both fixed and assured bandwidth had a service interval of 4 frames, i.e. 500 μsec, and the allocation bytes were chosen in such way that 64 kbps are assigned in a fixed manner and 38.08 Mbps are assured to the T-CONT.

To generate packets, all the ONUs had an application that created packets with exponentially distributed inter-arrival times. To vary the load, the mean inter-arrival time of the Poisson process was varied. We can see a summary of the parameters used in the simulation in Table I.

We can see the results of this simulation for average packet delay and lost packet ratio in Fig. 4 and Fig. 5, respectively. Here, we can see that using group-assured bandwidth improves the performance of the network, for the same amount of reserved capacity. E.g., considering the average delay, we can see a decrease of 7.5 to 17% when comparing GIANT and gGIANT with $n = 32$, depending on the load. This is because ONUs can now share their assured capacity when it is not needed by other elements of the same group. This is useful, even when the average throughput of the ONUs is the same, due to the burstiness of the traffic. From these figures, you can also see that increasing the group size, therefore the amount of bandwidth that is being shared, increases the benefits obtained from group-assured bandwidth.

While we can see benefits when the average load is equal, it is when the load is heterogeneous that bigger benefits can be achieved. To show this, we changed the load of only one ONU, while keeping the other ONUs at an average load of 28.08 Mbps. This meant that each T-CONT will have around 10 Mbps of unused capacity that can be shared with the group. We can see the average packet delay and packet loss ratio of these simulations in Fig. 6 and Fig. 7, respectively. Here, we can see that each time one ONU is added to the group, extra capacity is available to the ONU that needs it, thus allowing it to transmit 10 Mbps extra per ONU added to the group, before overloading the T-CONT. This can also be seen in the Figure 7 were we see that the ONU can transmit 10 extra Mbps per member added to the group before having a significant increase in lost packets.

Comparing these two scenarios, we can see that the results support the idea that gGIANT provides larger performance

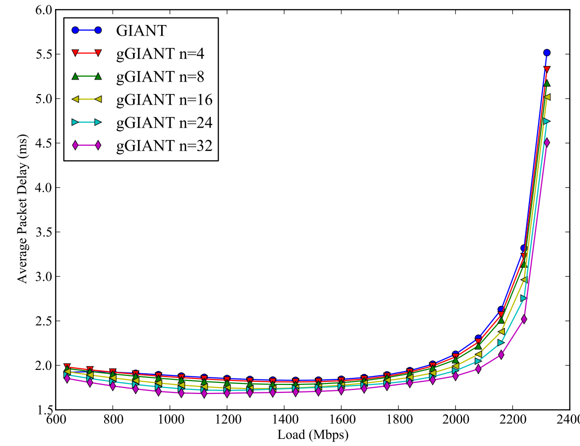


Fig. 4. Average Delay of upstream transmission when increasing the load of all ONUs

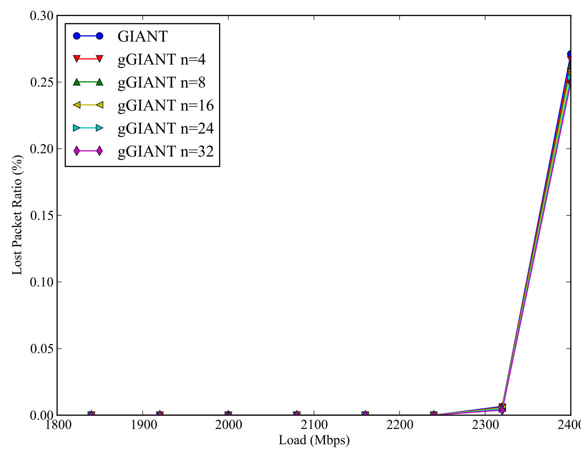


Fig. 5. Packets dropped when increasing the load of all ONUs

benefits when the traffic load is unbalanced, since there is more bandwidth available to share, indicating that group-assured bandwidth yields better results in case of a more realistic, heterogeneous traffic.

V. RESULTS OVER REAL MOBILE TRAFFIC TRACES

To assess the performance of group-assured bandwidth under a realistic scenario, we used the anonymised call-detail records kindly granted to us by an Irish mobile operator. These records hold information about mobile users' data transfers, such as session duration, transmitted bytes, base station location, etc., on multiple transmitters across Ireland, for both 2G and 3G technologies. Despite the great value of these traces, they have a few limitations. Firstly, they record only the initial sector where a data session is initiated, without any mobility information. To cope with this issue, we use the approximation that the entire session is carried out on the initial transmitter. Due to the short duration of the sessions we do not consider this to be a significant limitation [12]. The second limitation is that instantaneous throughput is not

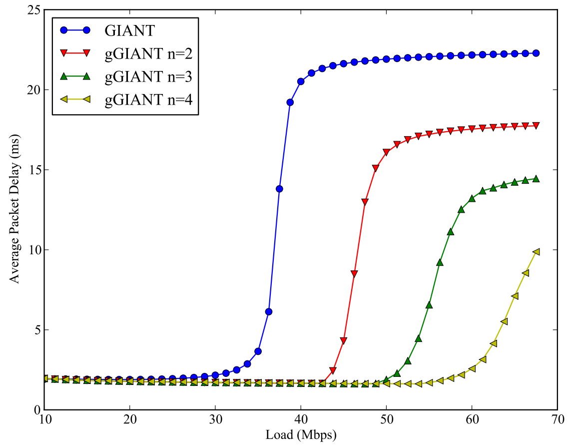


Fig. 6. Average Delay of upstream transmission when increasing the load of only one ONU.

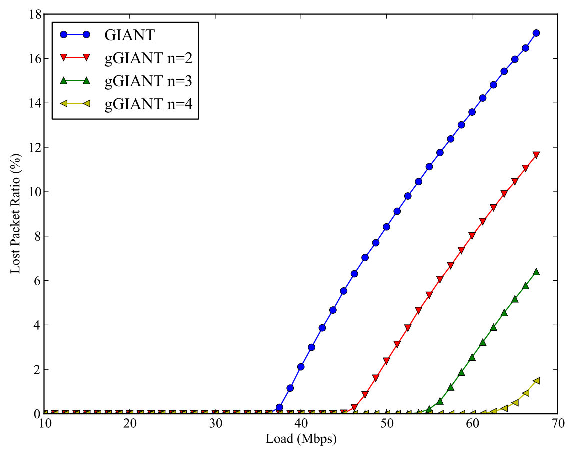


Fig. 7. Packets dropped when increasing the load of only one ONU.

recorded, only session durations and transmitted bytes, thus we can only approximate the load of each base station averaged over a certain duration of time.

Despite such limitations the traces have proven invaluable for our work as they provided information about load correlation between adjacent base stations. As seen in Section IV, the scheduler's performance improvement is greatly dependent of the discrepancies in load between base stations. Thus, having a realistic model for the traffic will give more realistic benefits compared to the unrealistic scenario of the homogeneous traffic. While this is not as ideal has having individual packet information, since it was still necessary to generate packets from a random distribution to evaluate our scheduling algorithm, this is a more realistic approach than either giving an homogeneous load to all base stations, or trying to guess the parameters of our random distribution. We then generated traffic by adopting an exponentially distributed packet arrival statistical model, using the average loads calculated by the traces. While this method does not assure the exact

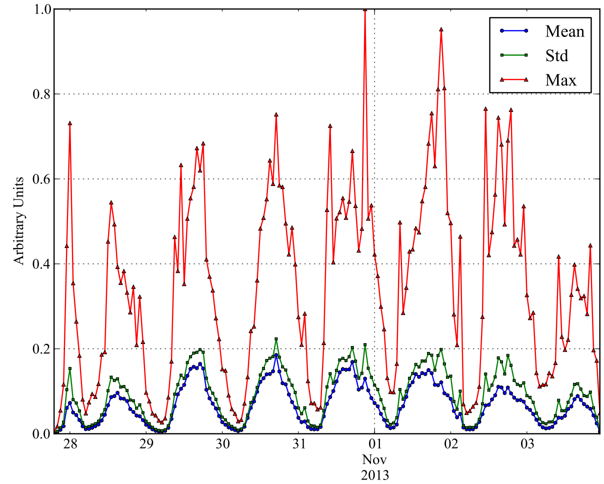


Fig. 8. Normalized mean, maximum and standard deviation of base stations' loads.

reproduction of backhaul traffic, it creates a traffic load for our packet-based simulator where the load correlation between base stations reflects that of a real mobile system.

In our analyses, we consider a week's worth of traffic from a small area in Dublin, that could be served by a PON. We consider that the traffic from multiple co-located transmitters is aggregated into a single ONU. This area is comprised of 31 different sites, each site with multiple transmitters. We can see the normalized maximum, mean and standard deviation of the network load in Fig. 8. We can also see in Fig. 8 that the variation of the load is considerable, with the standard deviation being similar to the mean. The load's variability is due to two factors: first, some locations of the city are more heavily loaded than others; second, some of the sites are composed just of 2G transmitters, thus having lower data rate, as compared to the sites composed of only 3G transmitters. This significant variability makes a good case for usage of group-assured bandwidth.

TABLE II
XG-PON SIMULATION PARAMETERS

Simulated time	50 seconds
Fibre propagation delay	0.4 msec
Number of ONUs	31
Service Interval	8 frames
Buffer Size	1 MBytes
Packet size (Bytes)	64 (60%), 500 (20%), 1500 (20%)
Inter-arrival times	Exponentially distributed

We carry out simulation over three scenarios: in scenario I, backhaul connections' assured capacity is provisioned to be equal to the average peak rate of the base stations; in scenario II, it is 30% below the average peak rate; in scenario III, it is 40% below the average peak rate. We summarize the most relevant simulation parameters in Table II.

The results showing the average packet delay are reported in Fig. 9, which is split into three sub-figures: Fig. 9a, Fig. 9b, and Fig. 9c representing scenarios I, II, and III respectively. This plot gives us three main insights. Firstly, if we assign capacity equal to the maximum, the GIANT algorithm still incurs in delay, due to the fluctuation of the packet arrival rate around the average peak value. Our gGIANT algorithm is instead able to eliminate such delay, because bytes that were reserved for particular T-CONTs, are now shared within the group, which allows T-CONTs to make their transmissions sooner when there is capacity available from the group. Secondly, even if we assign a capacity that is 30% less than the average peak, gGIANT is still able to keep the packet delay negligible, while with GIANT the delay greatly increases. Finally, we see that if the capacity is decreased to 40% of the average peak, even the gGIANT starts introducing significant delay, as there is not enough capacity to be redistributed.

Fig. 10 shows similar results, but considering packet loss rate rather than delay. Again, this figure is split into three subfigures: Fig. 10a, Fig. 10b, and Fig. 10c, representing scenario I, II, and III respectively. In this plot, we can see that when we assign capacity equal to the maximum, in GIANT there is some packet loss, due to fluctuations of packet arrival rate around the average peak value, while in gGIANT no packets are dropped since unused capacity is shared among the T-CONTs. We can see that when we decrease the assigned capacity by 30%, in GIANT the lost packet ratio increases significantly, up to 18%, while in gGIANT there are still no dropped packets. When the capacity is decreased by 40%, gGIANT starts to drop packets in some situations, but still maintains better performance than GIANT (7% maximum packet loss ratio against 22%). These results are consistent with those in Fig. 9.

VI. Conclusion

In this paper we investigated the benefits of group-assured bandwidth using artificial and real traffic traces for the use case of backhauling mobile base stations. Using real traffic traces, we were able to show that assuring bandwidth to groups of ONUs improves the performance while maintaining isolation from other users of the PON. Particularly, we showed that while using the same amount of assured bandwidth, having group assured bandwidth could reduce the networks average packet delay. It was also seen that when using group-assured bandwidth, it was possible to reserve 30% less capacity to the ONUs, while maintaining smaller average packet delays, when compared to the individual assured bandwidth case. This benefits stem from the properties of statistical multiplexing, since when using group-assured bandwidth, overloaded base stations can use spare capacity from other base stations.

Acknowledgment

This project has been part funded by the European Regional Development Fund through the Science Foundation of Ireland research centres programme under Grant Number 13/RC/2077 (CONNECT) and part funded by the European Union Seventh Framework Program (FP7/2007-2013) under grant agreement n. 318137 (Collaborative project DISCUS).

References

[1] M. Ruffini, L. Wosinska, M. Achouche, J. Chen, N. Doran, F. Farjady, J. Montalvo, P. Ossieur, B. O'Sullivan, N. Parsons, T. Pfeiffer, X.-Z. Qiu, C. Raack, H. Rohde, M. Schiano, P. Townsend, R. Wessaly, X. Yin, and D. Payne, "Discus: an end-to-end solution for ubiquitous broadband optical access," *Communications Magazine, IEEE*, vol. 52, no. 2, pp. S24–S32, February 2014.

[2] M. Ruffini, N. Doran, M. Achouche, N. Parsons, T. Pfeiffer, X. Yin, H. Rohde, M. Schiano, P. Ossieur, B. O'Sullivan, R. Wessaly, L. Wosinska, J. Montalvo, and D. Payne, "Discus: End-to-end network design for ubiquitous high speed broadband services," in *Transparent Optical Networks (ICTON), 2013 15th International Conference on*, June 2013, pp. 1–5.

[3] K. Kerpez, J. Cioffi, G. Ginis, M. Goldburg, S. Galli, and P. Silverman, "Software-defined access networks," *Communications Magazine, IEEE*, vol. 52, no. 9, pp. 152–159, September 2014.

[4] U. Paul, A. Subramanian, M. Buddhikot, and S. Das, "Understanding traffic dynamics in cellular data networks," in *INFOCOM, 2011 Proceedings IEEE*, April 2011, pp. 882–890.

[5] H. C. Leligou, C. Linardakis, K. Kanonakis, J. D. Angelopoulos, and T. Orphanoudakis, "Efficient medium arbitration of fsan-compliant gpons: Research articles," *Int. J. Commun. Syst.*, vol. 19, no. 5, pp. 603–617, Jun. 2006. [Online]. Available: http://dx.doi.org/10.1002/dac.v19:5

[6] P. Alvarez, N. Marchetti, D. Payne, and M. Ruffini, "Backhauling mobile systems with xg-pon using grouped assured bandwidth," in *Networks and Optical Communications - (NOC), 2014 19th European Conference on*, June 2014, pp. 91–96.

[7] J.-Y. Lee, I.-S. Hwang, A. Nikoukar, and A. Liem, "Comprehensive performance assessment of bipartition upstream bandwidth assignment schemes in gpon," *Optical Communications and Networking, IEEE/OSA Journal of*, vol. 5, no. 11, pp. 1285–1295, Nov 2013.

[8] M. S. Han, H. Yoo, and D. S. Lee, "Development of efficient dynamic bandwidth allocation algorithm for xgpon," *ETRI Journal*, vol. 35, no. 1, pp. 18–26, Feb. 2013.

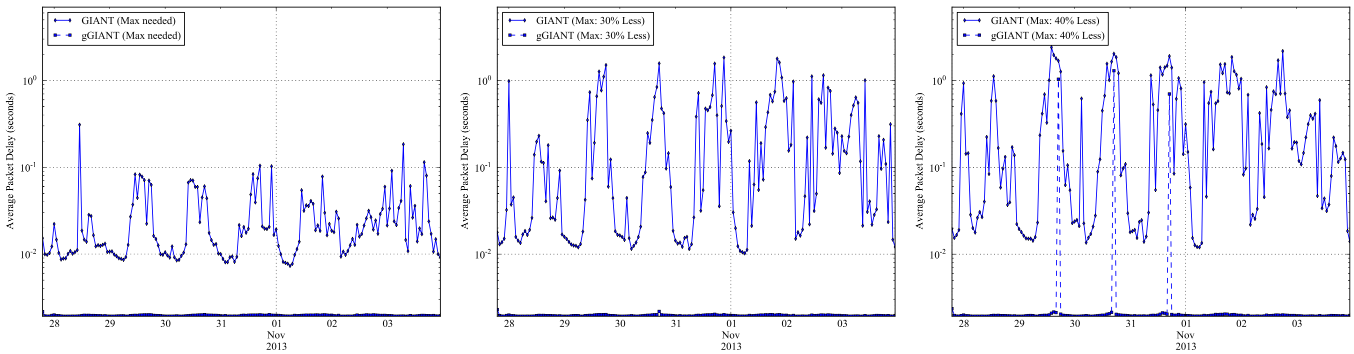
[9] M.-S. Han, "Simple and feasible dynamic bandwidth and polling allocation for xgpon," in *16th International Conference on Advanced Communication Technology (ICACT)*, Feb 2014, pp. 298–304.

[10] T. R. Henderson, M. Lacage, and G. F. Riley, "The ns-3 Network Simulator Modeling and Tools for Network Simulation," in *SIGCOMM demonstration*, 208.

[11] X. Wu, K. N. Brown, C. J. Sreenan, P. Alvarez, M. Ruffini, N. Marchetti, D. Payne, and L. Doyle, "An XG-PON Module for the ns-3 Network Simulator," in *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques (SimuTools)*, ICST, Brussels, Belgium, Belgium, 2013, pp. 195–202.

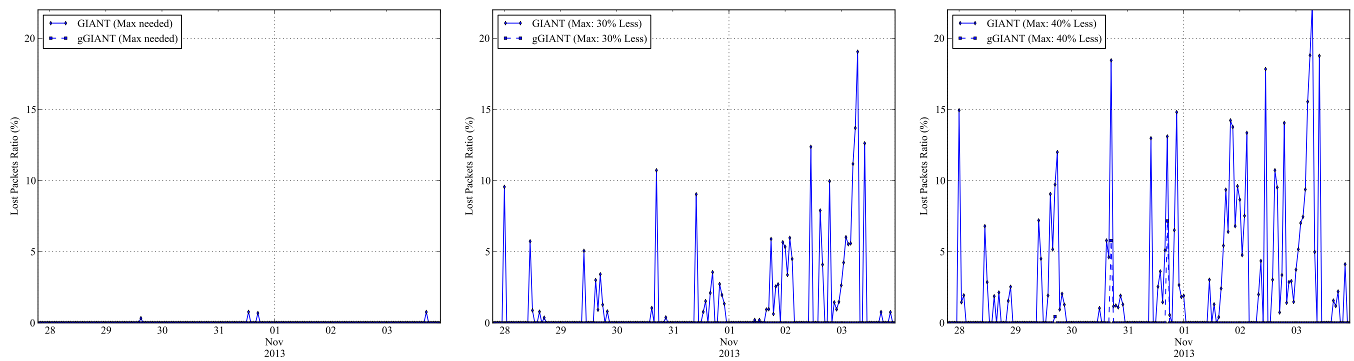
[12] D. Willkomm, S. Machiraju, J. Bolot, and A. Wolisz, "Primary users in cellular networks: A large-scale measurement study," in *New Frontiers in Dynamic Spectrum Access Networks, 2008. DySPAN 2008. 3rd IEEE Symposium on*, Oct 2008, pp. 1–11.

Pedro Alvarez Pedro Alvarez is a Ph.D. candidate at Trinity College Dublin, Ireland. He received a M.Sc. in Electrical and Computer Engineering from Instituto Superior Técnico, Portugal in 2011, where was distinguished with the Caixa Geral de Depósitos award. He has authored and co-authored 3 scientific publications and is one of the current developers of the XG-PON module for ns-3. His research focus on the integration of optical and wireless networks and on Dynamic Bandwidth Allocation algorithms for Passive Optical Networks.



(a) Maximum capacity needed assigned (b) 30% less capacity assigned (c) 40% less capacity assigned

Fig. 9. Network average packet delay for group and individual assured bandwidth.



(a) Maximum capacity needed assigned (b) 30% less capacity assigned (c) 40% less capacity assigned

Fig. 10. Network average packet loss for group and individual assured bandwidth.

Nicola Marchetti Dr. Nicola Marchetti is Assistant Professor in Wireless Communications at Trinity College Dublin, Ireland, and is a senior researcher of CONNECT / The Centre for Future Networks and Communications. He received the Ph.D. in Wireless Communications from Aalborg University, Denmark in 2007, and the M.Sc. in Electronic Engineering from University of Ferrara, Italy in 2003. He also holds a M.Sc. in Mathematics which he received from Aalborg University in 2010. He worked as a Research Assistant at the University of Ferrara in 2003-2004. He then was a Ph.D. Student in 2004-2007, and a Research and Teaching Post-Doc in 2007-2010 at Aalborg University. His former and current collaborations include research projects in cooperation with Samsung, Nokia Siemens Networks, Huawei, Intel Mobile Communications among others. His research interests include: 5G Wireless Communication Systems, Cognitive Radio and Dynamic Spectrum Access, Complex Systems Science, Integrated Optical-Wireless Networks, Massive MIMO Systems, Radio Resource Management, Ultra Dense Networks, and Waveforms. He authored in excess of 60 refereed journals and conference papers, holds 2 patents, and wrote 2 books and 4 book chapters.

Marco Ruffini Marco Ruffini received his M.Eng. in telecommunications in 2002 From Polytechnic University of Marche (Italy), and after working as a research scientist for Philips in Germany he joined Trinity College Dublin (TCD) in 2005, where he received his Ph.D. in 2007. Since 2010 is assistant professor (tenured 2014) at TCD. He is part of the CTVR/CONNECT telecommunications research centre at TCD and is currently technical coordinator of the FP7 DISCUS project and holder of a Science Foundation Ireland Principal Investigator award. He is author of more than 60 journal and conference publications and more than 10 patents. His research focuses on flexible high-capacity fiber broadband architectures.