Optimal virtual PON slicing to support ultra-low latency mesh traffic pattern in MEC-based Cloud-RAN

1st Sandip Das
CONNECT center
Trinity College Dublin
Dublin, Ireland
dassa@tcd.ie

2nd Marco Ruffini
CONNECT center
Trinity College Dublin
Dublin, Ireland
marco.ruffini@tcd.ie

***Abstract*—As progressive densification of cells, deployment of Cloud-RAN and Multi Access Edge Computing (MEC) are coming into reality to support the ultra-low latency with high reliability in 5G and beyond, it generates mesh traffic pattern across fronthaul network. This led to the evolution of PON architectural enhancements with virtualization in order to support such mesh traffic pattern. However, the allocation of virtual PON slices dynamically over such mesh-PON based fronthaul transport is becoming a research challenge. In this paper, we provide a mixed analytical-iterative model to compute optimal virtual PON slice allocation, providing mesh access connectivity with ultra-low end-to-end latency in next-generation MEC-based Cloud-RAN. Our proposed method can compute optimal virtual PON slice allocation in timescales compatible with real-time or near real-time operations.**

***Index Terms*—PON, MEC, Cloud-RAN, Low-Latency, Virtual PON, Slicing**

I. Introduction

Ultra-low end-to-end latency with high reliability is one of the key requirements in the next generation of networks for 5G and beyond. These requirements have led to architectural developments of communication networks centred around the use of Cloud-RAN, cell densification and the adoption of MEC nodes to reduce the distance between Radio Unit (RU) and the software processing site (Distributed Unit (DU), Central Unit (CU) up to the application layer). However, these new architectural designs generate mesh traffic communications patterns, with low latency and high capacity requirements, that are not easily (and cost-effectively) supported by current transport technologies. Current Passive Optical Network (PON) architectures only support point to multipoint traffic. A mesh topology could in principle be created through point-to-point links and packets switched technology (i.e., Ethernet), but this is far from cost-effective in high densification scenarios.

An emerging research approach to the problem is to modify existing PON architectures to support direct communication between endpoints. From early designs supporting fixed communication patterns across a small number of localised points [1], the approach has evolved towards higher flexibility [2] through a star coupler design, to finally improve scalability and programmability through dynamic virtual slicing [3]. At the same time, PON virtualisation has progressed from early models where schedulers were designed for specific services [4], to more agnostic algorithms targeting more generic performance indicators [5]. In addition to the traditional endpoint to central office (i.e., NORTH-SOUTH) communication, such architectures enable direct (i.e., EAST-WEST) communication between PON endpoints, which becomes a key feature to support high capacity and low latency interconnection of MEC nodes in the next-generation access network. The ONF AETHER [6] is a prominent example of this distributed access architecture, requiring a high-capacity, low latency mesh access network.

In this work, we address the dynamic network connectivity problem of an access network where a mesh PON topology enables dynamic interconnection of RUs, MEC nodes and central offices. The virtualized mesh topology can be created according to [7] where MEC nodes can be placed at the PON endpoints and EAST-WEST communication between PON end nodes can be established using Wavelength Loop Back (WLB) technique with reflective splitters. Here we are able to create virtual PON slices (whose capacity is allocated through the dynamic use of wavelength channels) to enable direct communications between RUs and MEC nodes (hosting DUs and possibly CUs) to support the operation of C-RAN instances.

This work proposes a method for the optimal formation of virtualised PON (vPON) slices under dynamic traffic scenarios. Given a number of RUs supporting a mix of 7.1 and 7.2 functional splits, with varying traffic load and pattern, we determine the optimal set of small cells, macro cells and MEC nodes (our virtual group of endpoints), that can support the required traffic while maintaining latency below a target threshold. Once the slices are created, our approach is also used to maintain the latency target, in real-time, below that threshold. As changes in traffic load and patterns produce latency increase above a threshold, we re-configure the virtual topology (i.e., MEC node migration) to reduce latency. A key achievement of this work is the development of an analytical

978-3-903176-33-1 © 2021 IFIP

model for PON latency, which significantly reduces the slice computation time, down to few tens of second (depending on load and number of iterations), which makes this algorithm suitable for real-time network optimisation.

II. System Model

Fig. 1, presents the system architecture and use case of the MESH-PON scenario where a Macro Cell with embedded MEC computation hosts an OLT, which enables direct communication to the nearby small cells (connected through ONUs). In this work, we assume such direct connectivity is achieved through Fibre Bragg Grating reflectors located at splitter locations (as with our previous work in [7]), although our solution is transparent to the specific physical layer implementation.

Fig 2 shows a sample solution returned by our virtual PON allocation model, with the minimum number of MEC nodes to guarantee latency below 100 μs threshold. Here, the macrocell and small cell coverage areas are modelled using the polybound-Voronoi diagrams. The small cells within the boundary of the corresponding macrocell (red colour borders) are connected by a level-1 PON tree with a possible MEC node (with OLT) deployed in the macrocell site. RUs at the small cells (blue dot) implement C-RAN with functional split 7.1 or 7.2 which is served by an ONU. Their OLT is located at the computing node that implements the corresponding DU (and possibly CU): this can either be a MEC node or a central office (depending on latency requirements). In addition, we use the common assumptions that MEC nodes are physically co-located with macro cell sites. The core network functions are hosted at the CO regardless of the placement of DU/CU.

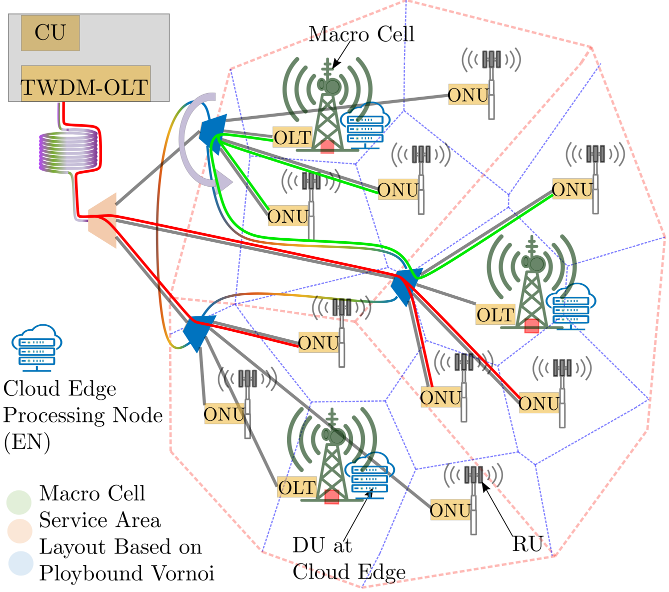

Fig. 1: Virtualised Mesh-PON architecture supporting MEC-based Cloud-RAN showing EAST-WEST links (green) along with traditional NORTH-SOUTH links (red).

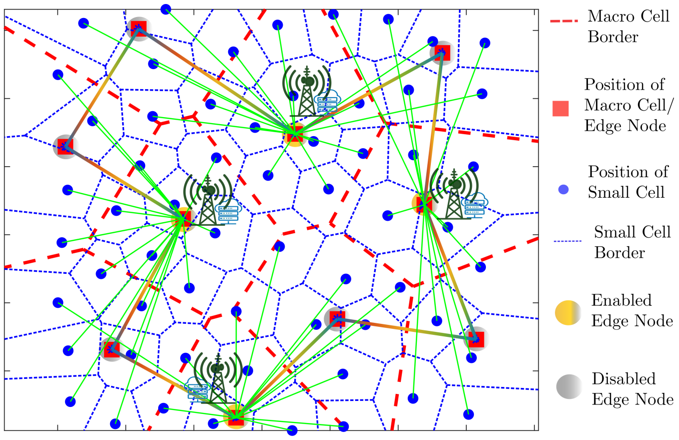

Fig. 2: Sample of network layout optimal solution computed by our mixed analytical-Iterative model. Only EAST-WEST links (green) are present due to the 100 μs latency constraint.

III. Optimal formation of vPON slice to provide ultra-low latency in uplink CoMP clusters

In order to find a solution for the virtual PON slice allocation problem, we first need to find an analytical expression for upstream latency per vPON slice based on parameters such as number of RUs, traffic at RUs and functional split (i.e., 7.1 or 7.2). To achieve this, we first find the packet queuing latency per vPON slice as a function of these parameters and then add the propagation latency, according to the slice configuration to obtain the end-to-end latency per vPON slice. We use the Kingsman heavy traffic approximation method for G/G/1 system for finding the mean packet queuing time in the ONU queue as described in (1) below.

$$T_W \leq \frac{1}{\mu C} + \frac{\lambda(\sigma_a^2 + \sigma_b^2)}{2(1-\rho)} \tag{1}$$

In (1), $1/\lambda$ is the average packet inter-arrival times, σ_a^2 is the variance of inter arrival times, σ_b^2 is variance of the service times, $\rho = \lambda/\mu$ is the utilisation factor and $1/\mu$ is the average service time per packet.

Let X be the random variable defining the service time of a fronthaul packet in uplink considering coordinated DBA for uplink packet scheduling. Let D be the set of possible fronthaul packet sizes arriving per Grant Cycle (GC) in a particular vPON slice, then the mean and the variance of the service times can be calculated as:

$$\begin{aligned} E\{X\} =& E\{k \times \Pr\{D_j = k\}\} \times D_{cap}^{UL} && (2)\\ 1/\mu =& E\{X\}, \quad \sigma_b^2 = E\{X^2\} - E\{X\})^2 && (3)\\ X =& \{X_k \,|\, X_k = k \times (1/D_{cap}),\ k \in D\} \end{aligned}$$

The state probability $Pr\{D = k\}$ (or $P(D)$) in (1), where random variable $D = \{\underline{x}_1, \underline{x}_2, \ldots, \underline{x}_N\}$ denotes the set of possible aggregated fronthaul packet size ($\underline{x}_i$) per uplink GC in a vPON slice can be found using successive convolution method [8] as given (4) and (5). Here, $P_j = \{p_j(d_i) \,|\, d_i \in$

$\{(d_1, d_2, \dots d_{K_{7.1}})|(d_1^{'}, d_2^{'}, \dots d_{K_{7.2}}^{'})\}\}$ are the state probabilities of fronthaul packet size for j^{th} RU considering two possible split configurations per RU, Split-7.1 or 7.2 and d_s is the eCPRI packet segment size.

$$P(\underline{x}) = P_1 \circledast P_2 \circledast \dots P_N, \quad D = \sum_{j=1}^{N} x_j.d_s \le d_{cap} \tag{4}$$

$$\begin{aligned} \text{Where,} \quad P_i \circledast P_j =& \{p_i(0).p_j(0), \sum_{x=0}^{1} p_i(x).p_j(1-x), \\ & \sum_{x=0}^{2} p_i(x).p_j(2-x), \dots, \sum_{x=0}^{u} p_i(x).p_j(u-x)\} \end{aligned} \tag{5}$$

$p_j(d_i)$ can be found by considering an M/M/m/m system (queuing theory) at the RU as follows.If γ and ν be the user call arrival and depart rate at each RU, then the probability that at the steady state, k-user is connected with RU is given by

$$\begin{aligned} p_k &= p_0 \left(\frac{\gamma}{\nu}\right)^k \frac{1}{k!} & k \le m \\ &= 0 & k > m \end{aligned} \tag{6}$$

$$\text{where} \quad p_0 = \left[\sum_{k=0}^{m} \left(\frac{\gamma}{\nu}\right)^k \frac{1}{k!}\right]^{-1}, \quad \text{and} \tag{7}$$

m is the maximum number of users that can be supported at the RU. If We consider thresholds $F_1 < F_2 < \dots < F_n$ as the number of active users per RU to jump between different eCPRI rates, then we can calculate the probability $p_j(d_i)$ that the fronthaul rate for RU-j is d_i as

$$\begin{aligned} & p_j(d_i) = \{F_{i-1} \le \text{number of users in the RU} \le F_i\} \\ \text{or} \quad & p_j(d_i) = \frac{\sum_{k=F_{i-1}}^{F_i} \left(\frac{\gamma}{\nu}\right)^k \frac{1}{k!}}{\sum_{k=0}^{m} \left(\frac{\gamma}{\nu}\right)^k \frac{1}{k!}} \end{aligned} \tag{8}$$

The next step is to define the model for optimal vPON slicing that satisfies an ultra-low latency threshold while minimising the total number of MEC nodes to be deployed for a given traffic intensity per RU. The decision variable $\alpha_i \in \{0,1\}$ determines whether a MEC-node$_i$ is to be deployed at the i^{th} level-1 PON tree or not. The objective function for the slice optimisation model is given in (9), and the constraints are described in (10)-(15). The optimal level-1 ring to minimize the uplink latency over EAST-WEST PON can be realised by getting a Hamiltonian tour for which the travelling distance is minimized, which is a classical Travelling salesman optimisation problem and can be formulated using (16) and (17),

Symbol	Description
$\mathcal{W}$	Set of wavelengths for EAST-WEST PON
r	RU-ID $r \in \{1, 2, \dots, \vert R \vert\}, (R :=$ No. of RUs)
v	Denotes the vPON ID $v \in \{1, 2, \dots, \vert V \vert\}$
α_i	Binary variable. 1 if an MEC-node, is to be deployed at the i^{th} level-1 PON tree.
Sl_v	Set of RUs belonging to vPON slice v i.e., $\{r \vert\ r \in R,$ and $X_{r,v} = 1\}$
$\mathcal{T}_{Sl_v,a}$	Theoretical value of uplink latency for vPON slice Sl_v and traffic load a (% load)
$\mathcal{T}_v^{max}$	Maximum value of uplink latency for vPON slice. Latency threshold (100 μs)
$\mathcal{L}_i$	Describes i^{th} physical Level-1 PON tree cluster. i.e., $\{r \vert\ r \in R$ and $X_{r,i} = 1\}$
$X_{r,v}$	Binary decision variable $\in \{0,1\}$. 1 if RU-r is assigned to vPON-v ($r \in R, v \in V$)
$\mathcal{K}_i^{\mathcal{W}}$	Set of $\mathcal{W}$-nearest neighbours of the i^{th} L1 PON tree.

TABLE I: Notations for mathematical symbols

$$\text{minimize:} \quad \sum_{i=1}^{N_{L_1}^{PON}} \alpha_i C_i^{CAP} + \sum_{i=1}^{N_{L_1}^{PON}} \alpha_i \mathcal{W} C_i^{OLT} \tag{9}$$

subject to (constraints):

$$\sum_{v \in V} X_{r,v} = 1 \quad \forall\ r \in R \tag{10}$$

$$\sum_{r \in R} X_{r,v} \le \alpha_v |R| \quad \forall\ v \in V \tag{11}$$

$$\sum_{r \in R} \sum_{v \in V} X_{r,v} = |R| \tag{12}$$

$$\mathcal{T}_{Sl_v,a} \le \mathcal{T}_v^{max} \quad \forall\ v \in V \tag{13}$$

$$\sum_{v \in K_v^{\mathcal{W}}} X_{r,v} = 1 \quad \forall\ r \in R \tag{14}$$

$$\sum_{i} \alpha_i = |V| \tag{15}$$

$$\text{minimize:} \quad \sum_{i \in 0,\dots,n-1} dist(e_i, e_{i+1}) + \ dist(e_n, e_0) \tag{16}$$

$$\text{subject to:} \qquad e_i \text{ are a permutation of } N \tag{17}$$

$$e_i := \text{edge Node Location}(< x_i, y_i > \ i \in \{1, 2 \dots N\})$$

The latency constraint given by (13) is a nonlinear function that can be solved with known nonlinear discrete optimisation solvers. However, the exhaustive search with such non-linear solvers is extremely slow due to the large search space and time-intensive non-linear constraint evaluation. Therefore, we propose an iterative optimisation method (Algorithm 1)(governed by parameter "Max iterations") along with integer-linear programming to take care of the non-linear constraint and speed up the optimisation significantly.

Algorithm 1: Iterative algorithm to solve the nonlinear discrete slice optimization model within bounded time

```
Set up and Initialize integer linear model $\mathcal{O}_{\mathcal{L}}$;
Solve $\mathcal{O}_{\mathcal{L}}$;
$N^{lb}_{MEC}$ = optimal no. of MEC nodes (output of $\mathcal{O}_{\mathcal{L}}$);
Evaluate non-linear constraint (13) for each vPON slice;
Find $\mathcal{N}$ := set of vPONs where non-linear constraint
  (13) is not satisfied ;
iteration_ID = 0;
while $\mathcal{N}$ is not empty do
    if max no. of iterations passed then
        increase lower bound: $N^{lb}_{MEC} = N^{lb}_{MEC} + 1$;
        iteration_ID=0;
    else
        iteration_ID=iteration_ID+1;
    end
    for each vPON slice $v \in \mathcal{N}$ do
        $X^{sol}_{r,v} \leftarrow$ current vPON config solution $X_{r,v}$;
        add constraints to $\mathcal{O}_{\mathcal{L}}$ :
            $\sum_{r \in R, X^{sol}_{r,v}=1} X_{r,v} \le \alpha_v(\sum_{r \in R} X^{sol}_{r,v} - 1)$;
    end
    solve updated $\mathcal{O}_{\mathcal{L}}$;
    if feasible/optimal solution obtained then
        Evaluate non-linear constraints (13);
        Find updated $\mathcal{N}$;
    else
        Result: No feasible solution. Exit
        ;
    end
end
Result: return the optimization output (slice config)
```

In a nutshell, this iterative method first evaluates the optimal no. of MEC nodes by evaluating the integer-linear programming model without the non-linear constraint ($\mathcal{O}_{\mathcal{L}}$). The obtained optimal no. of MEC nodes from the linear model is used as a lower bound (N^{lb}_{MEC}) for further exploration. Non-linear latency constraint is evaluated on solution obtained for each vPON slices (corresponding to MEC nodes) and checked for constraint violation. A slice configuration constraint is added for each of the latency violating slice of the linear-model solution, and the integer-linear model is run again. This process iterates until the non-linear latency constraint is passed for all the obtained slices. We define a maximum number of iterations to attempt to satisfy the non-linear constraint so that the algorithm is not stuck in the iteration for an indefinite time. Once the maximum number of iterations has passed for the current lower bound of the number of active MEC nodes, we increase the lower bound and start the process again until an optimal solution is found. Therefore, the max number of iterations creates a trade-off on the quality of the optimal solution vs the speed of the optimization.

IV. Performance Evaluation and Results

Our first step is the validation of the analytical model with simulations carried out in OMNET++. We consider a multi-wavelength architecture (i.e. following NG-PON2), although we assume a next-generation rate of 50Gbps per channel. The traffic from RU to DU is modeled as evolved Common Public Radio Interface (eCPRI) traffic. We consider split 7.1 and 7.2, both providing variable rate depending on the actual traffic at the cell. The corresponding fronthaul rates are derived from [9] and scaled to a 5G configuration of 100 MHz cell bandwidth. For an RU having four antennas and 4-MIMO layers, the fronthaul rate for split-7.1 goes from 1.378 to 7.384 Gb/s, while the split-7.2 goes from 273.98 Mbps to 2.92 Gbps. Our first result, in Fig 3, reports the feasibility region, showing the optimal mix of small cells using 7.1 and 7.2 split, that satisfies a given latency threshold (100 μs in this case). In Fig 3, different curves refer to different load at RUs, expressed as a percentage of average cell load. These results show how our analytical results based on queuing theory are in close agreement with simulations.

This is important because it means we can use the analytical model for calculating upstream latency to quickly find the optimal vPON slices through the proposed optimisation model instead of going through the extensive simulation for all possible vPON slice configurations for the considered network layout. It should be noticed that the solution to our optimisation problem also returns the specific MEC node location and virtual PON configuration (in variable $X_{r,v}$), a snapshot of which is shown in Fig. 2 ($X_{r,v}$ corresponds to the EAST-WEST green coloured links).

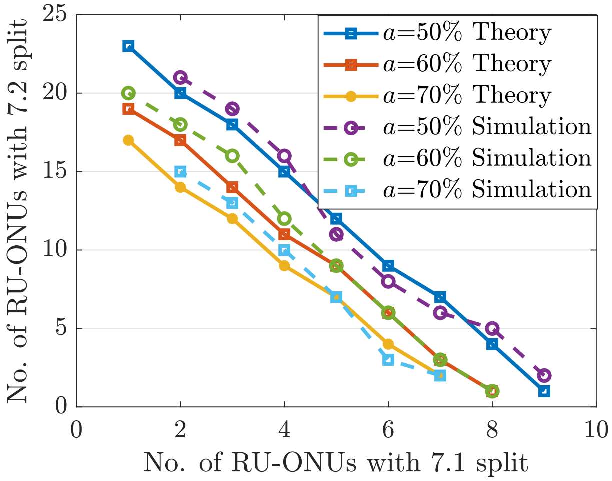

Fig. 3: Feasible vPON slice config. region: a is the RU percentage load.

In Fig. 4 and 5, we report the algorithm performance as a function of load and algorithm iterations. Fig. 4 shows how a higher number of iterations can improve the solution, returning a configuration with a smaller number of MEC nodes, as the solution is explored over a larger search space. We can also see that the computation time increases with the increase in traffic load. This is because, at a high traffic load, it is more difficult to find a solution that satisfies the latency constraint, therefore the algorithm spends more time in iteration to find the optimal values. Fig. 5 reports the exact computation time as a function of load and iterations.

From Fig. 4 and 5 we can conclude that our analytical-iterative model can quickly (i.e. within 10 iterations) find a solution suitable for real-time optimisation (i.e., following burst increase in RU load), which is close to optimal (form the figures, we see just one more MEC node compared to the higher iteration ones). At the same time, even the best solution can be calculated in times ranging from few seconds to few minutes. For comparison, we run the case of 30% load and 70 iterations in simulation without making use of the analytical model. While our model could return an optimal solution in about 100 seconds, the same result required 3 hours and 9 minutes without it (all computations were carried out on Intel i7-6600 mobile processor and simulations parallelised over 4 threads).

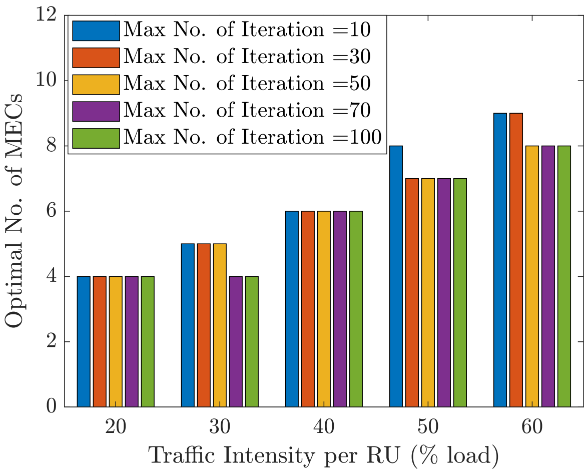

Fig. 4: Performance of solution for different algorithm iterations.

V. Conclusion

In this paper, we have proposed a mixed-analytical iterative optimization method that computes optimal virtual PON slice configuration in a mesh-PON type fronthaul network to support ultra-low latency under dynamic traffic scenarios. To achieve this, we first derived an analytical form for uplink latency in a virtual PON slice under varying traffic load, number of RUs in the vPON slice and RU-split configurations (7.1 or 7.2). With the help of discrete event simulation in OMNET++, we validated the analytical model. Using this analytical form of uplink latency, we then formed a non-linear discrete optimization framework to compute optimal virtual PON slices. We further proposed an iterative algorithm to solve the optimization model that can quickly find the optimal virtual PON slices with significantly reduced computation time. Our results show that using the proposed mix-analytical iterative optimization method, optimal virtual PON slices can be computed as quickly as seconds or tens of seconds (based on traffic load and max iterations). Thus making it suitable for real-time or near-real-time network optimization.

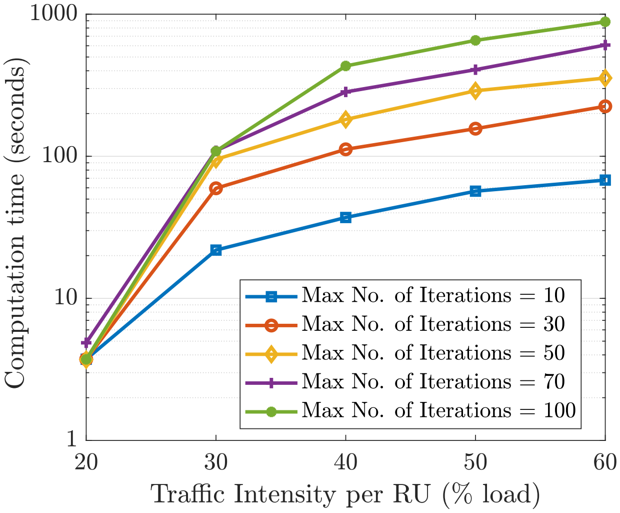

Fig. 5: Algorithm computation time vs load and iterations.

Acknowledgments

Financial support from Science Foundation Ireland grants 14/IA/2527 and 13/RC/2077 is gratefully acknowledged.

References

[1] T. Pfeiffer. “Converged heterogeneous optical metro-access networks,” Tu.5.B.1, OFC 2010.

[2] C. Ranaweera et al., “Next-generation optical-wireless converged network architectures,” IEEE Network, vol. 26, no. 2, pp. 22–27, Mar. 2012.

[3] S. Das, M. Ruffini. “PON Virtualisation with EAST-WEST Communications for Low-Latency Converged Multi-Access Edge Computing (MEC)”, M2H.3., OFC 2020.

[4] P. Alvarez et al., ”Backhauling mobile systems with XG-PON using grouped assured bandwidth,” NOC 2014.

[5] M. Ruffini et al., ”The Virtual DBA: Virtualizing Passive Optical Networks to Enable Multi-Service Operation in True Multi-Tenant Environments—. OSA Journal of Optical Communications and Networking, No.4, Vol.12, April 2020.

[6] Aether: Enabling a New Era of Smart Enterprises. ONF Whitepaper, December 2020.

[7] S. Das et al., “Virtualized EAST–WEST PON architecture supporting low-latency communication for mobile functional split based on multi-access edge computing,” OSA Journal of Optical Communications and Networking, vol. 12, no. 10, pp. D109-D119, October 2020.

[8] V. B. Iversen, “Teletraffic engineering and Network planning, Chap-10, Sec-4.”, DTU Fotonik, 2007.

[9] U. Dotsch et al., “Quantitative analysis of split base station processing and determination of advantageous architectures for LTE”, Bell Labs Technical Journal, vol. 18, no. 1, pp. 105–128, Jun. 2013.