Performance Evaluation of QoS-Aware Layer-2 VPNs over Fiber-Wireless (FiWi) Networks

Ahmad R. Dhaini*, Pin-Han Ho*, Abdallah Shami†, and Xiaohong Jiang‡
*Department of Electrical and Computer Engineering, University of Waterloo
†Department of Electrical and Computer Engineering, University of Western Ontario
‡Graduate School of Information Sciences, Tohoku University

***Abstract*—The integration of Ethernet Passive Optical Networks (EPONs) and IEEE 802.16 (WIMAX) has been lately presented as a promising candidate for deploying fiber-wireless (FiWi) broadband networks. Conversely, lightweight layer-2 virtual private networks (VPNs), which can provide bandwidth guarantee to the respective users, were only addressed in [1] in the context of the fixed-mobile convergence (FMC). In this paper, we present a variation of the QoS-provioning framework of [1]. Here, the upstream bandwidth is distributed among VPNs in a finer way, in order to efficiently utilize the network resources. We also present a generic analytical model to evaluate the performance of each registered VPN service. Our proposed model applies for wireless and optical domains and provides performance measurements such as packet queueing delay, end-to-end (from wireless user to optical server) packet delay and average queue size. Numerical results are compared with simulation experiments, and show consistency between both outcomes.**

I. Introduction

The integration of Ethernet Passive Optical Network (EPON) and WIMAX has been lately presented as an attractive solution for realizing fixed mobile convergence (FMC) [2], [3]. The bandwidth benefit of fiber communications and the mobile non-line-of-sight (NLOS) features of wireless communications, in addition to the fact that both EPON and WIMAX belong to the same standardization body (IEEE 802.x), make the deployment of such a network smooth and simple.

Virtual private networks (VPNs) over EPON-WIMAX were firstly introduced in [1]. Such VPNs are referred to as *layer-2 VPNs* in the sense that they are built upon the medium access control (MAC) layer protocol. These VPNs allow for the support of premium services with custom-designed control, dedicated connectivity, diverse quality-of-service (QoS) requirements and security assurance [4]; features that are essential for private and/or mission-critical systems and services. Particularly, building up layer-2 VPNs is considered the best suitable when an EPON-WIMAX integrated network is deployed, as opposed to using layer-3 and/or layer-1. This is due to 1) the physical layer heterogeneity of EPON and WIMAX and 2) the different network dynamics and fast channel changing status. Nonetheless, supporting layer-2 VPNs over EPON-WIMAX entitles the resolution of many challenges such as, but not limited to, resource allocation, admission control, routing and VPN configuration and management.

In [1], a new framework, namely WIMAX-V**P**ON, was presented to address the resource management and admission control problems. More specifically, the proposed framework consisted of a new joint VPN-based admission control (AC) and an uplink/upstream dynamic bandwidth allocation (DBA) paradigm that ensures bandwidth guarantee for each VPN service. Simulations results proved the effectiveness of the framework. However, the QoS-provisioning component of the framework did not take into account the control overhead caused by the polling mechanism adopted in both EPON and WIMAX, when distributing the upstream bandwidth among the registered VPNs. Furthermore, no analytical model was provided to verify the results numerically. For these reasons in this paper, we present a variation of the QoS-provisonning component to recognize the aforementioned control overhead, which enables a finer bandwidth distribution. This variation will then help us design a generic analytical model to perform an approximative study on the performance of the supported VPN services over the proposed framework, in both the wireless and optical domains. Using classical queueing theory, our analytical model is able to provide a fine estimation of the network behavior with different input parameters. Such a study can greatly help in terms of estimating the network resilience against various VPN services with diverse QoS requirements; in order to ultimately set the proper network parameters, as well as to perform some optimization, if needed.

The rest of the paper is organized as follows. A brief overview of WIMAX-V**P**ON along with the new QoS-provisioning component is presented in Section II. The proposed analytical model is shown in Section III. Section IV presents numerical results and we conclude in Section V.

II. WIMAX-V**P**ON: A brief overview

In WIMAX-V**P**ON [1], each layer-2 VPN serves as a *shim layer* that maps the VPN service requirements and commands to the MAC layer, via a suite of service access points (SAPs) and primitives. Thus, each VPN corresponds to a specific service requirement bundle allowing the users to dynamically configure their service requirements. This feature is essential to support stringent bandwidth guarantee and possible preemption requests. Nonetheless, such diversified VPN requirements require MAC paradigms and protocols to be mapped to the underlying EPON-WIMAX architecture, in order to ensure a statistical QoS performance guarantee. In that context, we next present the new QoS-privioning component (a variant of [1]), and we overview the joint VPN-based AC/DBA designed to complement the framework.

A. VPN-based QoS provisioning

With the new QoS-provisioning variant (Fig. 1), the effective upstream V**P**ON cycle T_{eff}^{VPON}, which is the upstream optical polling interval (PI) length minus the control overhead that is caused by the polling and requesting signaling, is divided into two *sub-cycles*. The first sub-cycle βT_{eff}^{VPON} is shared among all the K VPNs. The second sub-cycle $(1-\beta)T_{eff}^{VPON}$ is shared among non-VPN services. Note that the T_{eff}^{VPON} can either be obtained via simulations by measuring the maximum throughput over the transmission rate [1], or analytically (as we will see next). Let B_{min}^{k} be the bandwidth reserved for VPN k (denoted as V_k) in each PI, and R_N the transmission speed of PON in Mbps. In addition, let each V_k be given a weight w_k to determine its *paid/committed* bandwidth. Therefore, B_{min}^{k} (in bytes, therefore divide by 8) can be computed as follows:

$$B_{min}^{k} = \frac{\beta T_{eff}^{VPON} \times R_N \times w_k}{8}, \text{where} \left(\sum_{k=1}^{K} w_k = 1\right) \quad (1)$$

To free best-effort (BE) traffic from starvation, we reserve it a quota of αB_{min}^{k}, while the real-time flows will share the remaining bandwidth, that is $(1-\alpha)B_{min}^{k}$ [1].

B. A Joint VPN-based AC & DBA Scheme

The proposed VPN-based admission control (VPN-AC [1]) admits an incoming constant-bit-rate (CBR, e.g., UGS) flow if its mean rate can be accommodated in both the wireless total capacity and VPN bandwidth share. On the other hand for variable-bit-rate (VBR, e.g., rtPS and nrtPS) traffic, a *guaranteed rate* (with specific delay requirements) is extracted from the arrival process passing through the dual-token leaky bucket (DTLB) that is situated at the MAC buffer entrance. Consequently, a VBR flow is admitted if its guaranteed rate can be accommodated in the network [5].

The proposed VPN-DBA [1] mainly divides each upstream/uplink cycle/frame into two sub-cycles. The first sub-cycle is used to allocate *guaranteed bandwidth* for admitted real-time traffic, whereas the second sub-cycle is used to allocate BE traffic per VPN. Hence, the second sub-cycle is also divided into smaller VPN-BE sub-cycles. Moreover at the ONU-BS, VPN-DBA accounts for different channel conditions of each subscriber station (SS), by having the allocated time share be adaptive to the fluctuating channel, such that a QoS statistical bandwidth guarantee is achieved.

The communication between the OLT/ONU-BS and the ONU-BSs/SSs is polling-based. That is, each user is polled in one PI, requests bandwidth, and gets assigned a transmission slot in the next PI. The access technology adopted is OFDM-TDMA in the wireless domain and TDMA in the optical one.

III. Analytical Model

In our model, we refer to the polling node as *server*, and the polled station as *client*. That is, the OLT is considered a server when polling the ONU-BS (i.e., client); whereas the ONU-BS acts as a server when polling an SS (i.e., client). We also define the following notations:

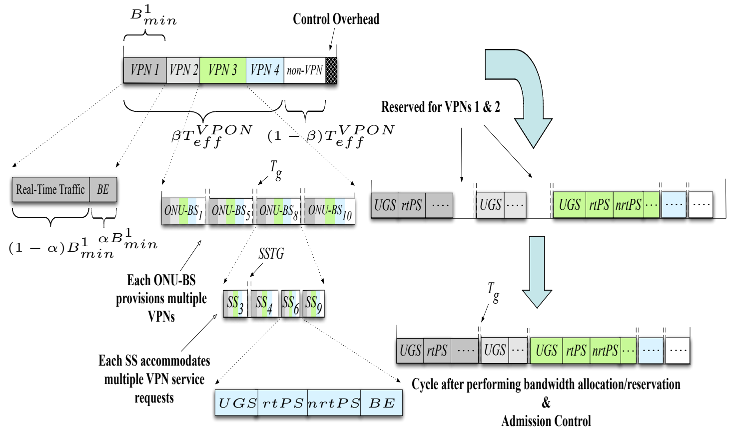


Fig. 1: Proposed V**P**ON QoS-Provisionning Variant

T_{ir}	Initial Ranging Period
T_{up}	Uplink/Upstream frame/cycle length
T_{dl}	Downlink frame length
T_{prop}	Message propagation time (DL or UL)
T_{dba}	Server's DBA computation time
T_{gr}^{proc}	Grant message processing time
T_{gr}^{tran}	Grant message transmission time
T_g	Guard time or Preamble
T_{br}	Bandwidth request message transmission time

We assume that the arrival processes of all clients are Poisson-distributed and identical and independently distributed (i.i.d.), with per-client i average rate $\lambda_i = \sum_{c=1}^{n-1} \lambda_c$. Here, it is important to note that we are aware that the arrival process of VBR traffic (or the guaranteed rate), that is injected in the network after passing through the DTLB, is not poisson distributed. However as mentioned, the purpose of our model is to provide a fine approximation of the estimated network behavior; rather than claiming an exact one. In that sense, we also use Poisson to model the guaranteed rate, due to the rich properties that Poisson traffic possess, and that will facilitate our analysis. Our numerical results will later show that this assumption is statistically adequate. We also apply the following relevant assumptions:

1) The wireless channel quality of each connection remains constant on a per-frame basis.
2) The buffer size of each queue is large enough to accomodate the backlogged traffic, i.e., no packet dropping.
3) Clients (ONU-BSs and SSs) are separated from the server (OLT and ONU-BS) with the same distance.the upper bound of the VBR traffic delay.
4) We assume that a traffic of any class c is the same for all VPNs. That is, $\forall i, \lambda_{c,i}^{1} = \lambda_{c,i}^{k} = \cdots = \lambda_{c,i}^{K}; (c = 1, \ldots, n)$ and $(k = 1, \ldots, K)$; where $\lambda_{c,i}$ is the arrival rate per class c at each client i.

As noticed, our network is made out of multiple clients, with each client having multiple queues, accessing the channel in a TDMA fashion (or OFDM-TDMA in the wireless domain). Each client is assumed to have a scheduler to select packets from each priority queue according to the bandwidth allocated by the server. Hence, each client can be modeled as a weighted round robin (WRR) scheduler, but with one

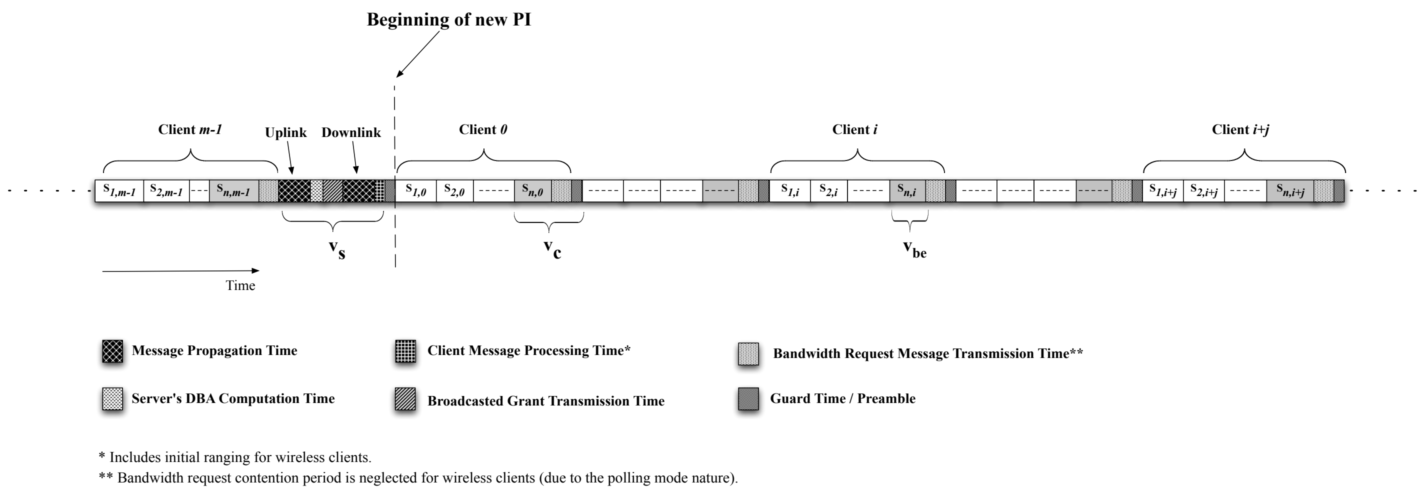


Fig. 2: Illustration of temporal events in each Polling Interval (PI)

round of scheduling only. The number of packets that are transmitted in this round are equal to the arrival (or guaranteed) rate of each service. Nonetheless due to the TDMA channel access, each scheduler will be exposed to a "vacation" time before accessing its portion of channel. As a result, our analysis is basically the combination of WRR, with one round scheduling, and M/G/1 queueing with vacations [6], [7]. The vacation periods are divided into two folds: server vacation and client vacation. A server vacation, denoted v_s, is the idle period displayed in the system between two consecutive PIs; whereas a client vacation, denoted v_c, is the idle period that is exhibited between two consecutive transmission time slots. A detailed graphical representation of the temporal events with the respective vacations, is depicted in Fig. 2. Additionally, the purpose of our model is to verify analytically that the QoS requirements of real-time VPN services are met. Therefore, we only derive the expected delay and queue size for real-time queues (i.e., $c = 1, \ldots, n-1$); where class n is for BE traffic. Nonetheless, because VPN-DBA/AC accommodates BE traffic by reserving a minimum per-VPN BE throughput rate, we may want to test the performance of real-time VPN services with the existence of heavy-load BE traffic (i.e., the reserved rate is fully utilized). To implement the latter, we assume that the total BE (of all VPNs) reserved rate (or translated to period of time) portion is included in the vacation times for real-time queues of each client. More specifically, we assume that there exists at least one user per V_k, and this user may fully access the reserved BE bandwidth for V_k. Let m_o and m_w be the number of optical clients connecting to the optical server, and number of wireless clients connecting to one wireless server, respectively. Using Eq. (1), we can compute the average per-client BE vacation time $\overline{v}_{BE}$ as following:

$$\overline{v}_{be} = \begin{cases} \dfrac{\sum_{k=1}^{K} \alpha^k B_{min}^k}{m_o \times R_N} & \text{(Fi)} \\ \\ \dfrac{\sum_{k=1}^{K} \alpha^k B_{min}^k}{m_o \times m_w \times \overline{R_x}} & \text{(Wi)} \end{cases} \tag{2}$$

where, $\overline{R_x} = (\sum_{i=0}^{m_w-1} R_i^x)/m_w$, defined as the average transmission rate in one wireless domain (i.e., at one server). The vacation intervals v_c and v_s of each client are computed based on the domain the client belongs to (Fi) or (Wi). v_c is uniform in both domains and is given by:

$$v_c = \overline{v}_{be} + T_{br} + T_g \tag{3}$$

While v_c is uniform, v_s varies due the TDD nature of WIMAX. In addition, since we consider polling-based services, we exclude the contention period (for wireless clients) from our calculations, and compue v_s as follows:

$$v_s = \begin{cases} 2T_{prop} + T_{dba} + T_{gr}^{proc} + T_{gr}^{tran} & \text{(Fi)} \\ \\ T_{dl} + T_{ir} + 2T_{prop} + T_{dba} + T_{gr}^{proc} + T_{gr}^{tran} & \text{(Wi)} \end{cases} \tag{4}$$

We also define $AI_i(t)$ as the access interval (vacation and data) for a client i in interval $[0, t)$ such that:

$$AI_i(t) = v_i + \sum_{c=1}^{n-1} S_{c,i}(t) \tag{5}$$

where $S_{c,i}(t)$ is the number of packets serviced for class c of client i, and v_i is the vacation period appearing just before i's data interval. As shown in Fig 2, v_i depends on i's position in each PI and is estimated as follows:

$$v_i = \begin{cases} v_c + v_s & (i = 0) \\ v_c & (1 \leq i \leq m-1) \end{cases} \tag{6}$$

In the optical domain, the average arrival rate at class c of a wireless server is the sum of all arrival rates of class c of all wireless clients connecting to the same server. Hence, the total arrival rate λ will then be calculated in the following manner:

$$\lambda = \begin{cases} \displaystyle\sum_{j=0}^{m_o-1} \sum_{i=0}^{m_w-1} \sum_{c=1}^{n-1} \lambda_{c,i,j} = m_w m_o \sum_{c=1}^{n-1} \lambda_c & \text{(Fi)} \\ \\ \displaystyle\sum_{i=0}^{m_w-1} \sum_{c=1}^{n-1} \lambda_{c,i} = m_w \sum_{c=1}^{n-1} \lambda_c & \text{(Wi)} \end{cases} \tag{7}$$

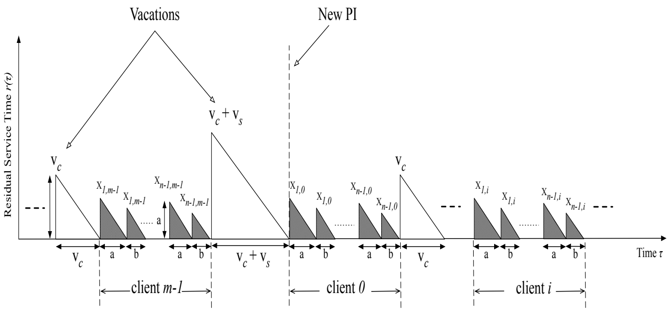


Fig. 3: Derivation of the mean residual time $\overline{R_i}$ during $[0, t)$

The first and second moments of the packet transmission times belonging to class c are $E[X_c] = \overline{X_c} = 1/\mu_c$, and $E[X_c^2] = \overline{X_c^2}$, respectively. Since we assume that all traffic of the same class for different VPNs are equal, and for simplicity, we omit the usage of k in our notations. In addition, in the derivations that will follow, we use m to represent the number of clients connected to one server. The utilization factor per class c is $\rho_c = \lambda_c \overline{X}_c$. We also assume that a steady state always exists. That is,

$$\rho_1 + \rho_2 + \cdots + \rho_{n-1} < 1 \tag{8}$$

Inter-arrival times and transmission times are, as usual, assumed independent. While we assume that all clients have identical arrival and service statistics, we allow the reservation intervals of different clients to have different statistics. Our model is also concerned about measuring the following classes: UGS, which has the highest priority and is modeled as a CBR traffic; followed by rtPS then nrtPS which are both modeled as VBR traffic. In addition, we assume that a packet may arrive to any client (to any queue), during the whole PI.

A. End-to-End (E2E) Packet Delay

Each incoming packet ξ belonging to any VPN service of class c, for any domain (Fi or Wi), will be subject to a total delay D_c, such that:

$$D_c(\xi) = W_c(\xi) + T_{prop} + X_c(\xi) \tag{9}$$

where $W_c(\xi)$ and $X_c(\xi)$ are ξ's average queueing delay and transmission time, respectively. At steady state, as $\xi \to \infty$, the expected value of the total delay is then given as:

$$E\{D_c\} = E\{W_c\} + T_{prop} + E\{X_c\} \tag{10}$$

$E\{X_c\}$ is given by $1/\mu_c$ and $T_{prop} = distance/speed$. Therefore in order to compute $E\{D_c\}$ (or $\overline{D_c}$), we need to derive an expression for $E\{W_c\}$.

Once computed, the end-to-end (E2E) packet delay (i.e., from the wireless client to the optical server) will then be equal to the following:

$$\overline{D_c}(\text{E2E}) = \overline{D_c}(\text{Fi}) + \overline{D_c}(\text{Wi}) \tag{11}$$

B. Queuing Delay Analysis

To compute queueing delay of packet (ξ) arriving to any class $c : 1 \to n-1$, of client i, we first denote:

- W_i The waiting time of any packet in queue of client i.
- R_i The residual service time seen by a packet arriving to client i.
- N_i The number of packets, of client i, waiting to be transmitted before ξ. This shall include all the packets from all priority queues.
- Y_i The reservation/gap interval seen by a packet arriving to client i before it is transmitted.

The queueing delay of ξ arriving at client i is given by:

$$W_i(\xi) = R_i(\xi) + N_i(\xi) \times \overline{X}(\xi) + Y_i(\xi) \tag{12}$$

where $\overline{X}(\xi)$ is the service time of packet ξ. We note that $E\{R_i\} = \lim_{\xi \to \infty} R_i(\xi)$. Similarly, $E\{Y_i\} = \lim_{\xi \to \infty} Y_i(\xi)$, $E\{N_i\} = \lim_{\xi \to \infty} N_i(\xi)$ and $E\{W_i\} = \lim_{\xi \to \infty} W_i(\xi)$.

We first compute the queueing delay for classes 1 and 2, and then provide a generalized expression for class c, where $2 < c \leq n-1$. If ξ belongs to class 1, its expected queueing delay will then be equal to the following:

$$E\{W_{1,i}\} = \frac{E\{R_i\}}{\psi_i} + \frac{E\{N_{1,i}\}}{\mu_c} + E\{Y_i\} \tag{13}$$

where, ψ_i is client i's AI proportional to all clients' AIs during $[0, t]$:

$$\psi_i \triangleq \lim_{t \to \infty} \left(\frac{AI_i(t)}{\sum_{j=0}^{m-1} AI_j(t)} \right) \tag{14}$$

In steady state, ξ will see the same average number of packets queued, at both its enqueuing and dequeuing times.The expected number of arriving packets to class 1, at those times, is equivalent to $\lambda_1 E\{W_1\}$. As a result:

$$\begin{aligned} \frac{E\{N_{1,i}\}}{\mu_1} &= \lambda_1 E\{W_1\} \times \overline{X_{c,i}} \\ &= \rho_1 E\{W_1\} \end{aligned} \tag{15}$$

Hence, the expected delay of the priority queue 1 is then computed as follows:

$$\begin{aligned} E\{W_{1,i}\} &= \frac{(E\{R_i\}/\psi_i) + E\{Y_i\}}{1-\rho_1} \\ &= \frac{(\overline{R_i}/\psi_i) + \overline{Y_i}}{1-\rho_1} \end{aligned} \tag{16}$$

According to the aforementioned client i's WRR (with one round) scheduling mechansism, the expected delay of a packet arriving at the second queue shall wait for all the packets in the higher priority queue and the packets that arrived earlier to the same priority queue, in addition to $\overline{R_i}$ and $\overline{Y_i}$:

$$\begin{aligned} E\{W_{2,i}\} &= \frac{E\{R_i\}}{\psi_i} + \frac{E\{N_{1,i}\}}{\mu_1} + \frac{E\{N_{2,i}\}}{\mu_2} + E\{Y_i\} \\ &= \frac{(\overline{R_i}/\psi_i) + \overline{Y_i}}{(1-\rho_1)(1-\rho_2)} \end{aligned} \tag{17}$$

In the same manner, a general expression of the expected queueing delay for class c, where $2 < c \leq n-1$, can then be given by:

$$E\{W_{c,i}\} = \frac{(\overline{R_i}/\psi_i) + \overline{Y_i}}{(1-\rho_1)(1-\rho_2)\ldots(1-\rho_c)} \tag{18}$$

To find the value of $\overline{R_i}$, we use the concept of the *mean residual service time* [6], [7] with a graphical argument. As shown in Fig. 3, $r_i(\tau)$ (the residual time at time τ) exhibits vacation times depending on the position of client i in each PI. Therefore, the time average of $r_i(\tau)$ in $[0,t]$ is:

$$\frac{1}{t}\int_0^t r_i(\tau)d\tau = \frac{1}{t}\left(\sum_{x_i\in S_i(t)}\frac{1}{2}x_i^2 + \sum_{v_i\in V_i(t)}\frac{1}{2}v_i^2\right) \quad (19)$$

where $V_i(t)$ is the number of vacations appeared in AI_i during time interval $[0,t]$. Since it is assumed that all clients are equally loaded, the number of vacations appeared in AI_i during $[0,t]$ can be evaluated as just $V(t)/m$, with $V(t)$ as the total number of vacations occurred during $[0,t]$; as these vacations appearing in AI_i repeat once in every m consecutive vacations. Denoting $M_{c,i}(t)$ as the number of class c packets serviced by client i during $[0,t]$, we can rewrite Eq. (19) in the following way:

$$\overline{R_i} = \frac{1}{2}\left[\Big(\sum_{c=1}^{n-1}\frac{M_{c,i}(t)}{t}\Big)\overline{X_i^2} + \frac{1}{m}\Big(\frac{V(t)}{t}\times v_i^2\Big)\right]$$

where v_i is computed from Eq. (6), and $\overline{X_i^2}$ denotes the second moment of client i's service time averaged over all classes. More specifically,

$$\overline{X_i^2} = \frac{\lambda_{1,i}}{\sum_{c=1}^{n-1}\lambda_{c,i}}\overline{X_{1,i}^2} + \cdots + \frac{\lambda_{n-1,i}}{\sum_{c=1}^{n-1}\lambda_{c,i}}\overline{X_{n-1,i}^2}$$

We note that $\lim_{t\to\infty}\sum_{c=1}^{n-1}\frac{M_{c,i}(t)}{t} = \sum_{c=1}^{n-1}\lambda_{c,i}$. Let $\varphi_{n-1} = \sum_{c=1}^{n-1}\rho_c$. As a result, $\lim_{t\to\infty}V(t) = t(1-\varphi_{n-1})/\overline{v}$, where $\overline{v} = (mv_c+v_s)/m$ is the average duration of one vacation period. When $t\to\infty$, the mean residual time of client i is then given as:

$$\overline{R_i} = \frac{1}{2}\left[\Big(\sum_{c=1}^{n-1}\lambda_{c,i}\overline{X_{c,i}^2}\Big) + \frac{v_i^2(1-\varphi_{n-1})}{mv_c+v_s}\right] \quad (20)$$

The value of ψ_i in Eq. (21), is basically equivalent to the computation of the probability that packet ξ arrives during $AI_i(t)$ when $t\to\infty$, denoted as $\pi_i^\xi(t)$. Since all clients have equal data rates, the data intervals of all clients have equal average length at steady state. As a result at steady state, ξ will arrive during client i's data interval with probability φ_{n-1}, and to its vacation interval with probability $(1-\varphi_{n-1})v_i/\sum_{j=0}^{m-1}\overline{v}_j$ [6]:

$$\psi_i = \lim_{t\to\infty}\pi_i^\xi(t) = \frac{\varphi_{n-1}}{m} + (1-\varphi_{n-1})\frac{v_i}{\sum_{j=0}^{m-1}\overline{v}_j}$$
$$= \begin{cases}\dfrac{\varphi_{n-1}}{m} + (1-\varphi_{n-1})\dfrac{v_c+v_s}{mv_c+v_s} & (i=0)\\[2ex] \dfrac{\varphi_{n-1}}{m} + (1-\varphi_{n-1})\dfrac{v_c}{mv_c+v_s} & (1\le i\le m-1)\end{cases} \quad (21)$$

To determine Y_i, we first denote:

$$\gamma_{ij} = E\{Y_i(\xi)|\pi_{i,j}^\xi\}$$

where, $\pi_{i,j}^\xi(t) = P\Big\{$packet ξ arrives during $AI_i(t)$ and belongs to client $(i+j)\%m\Big\}$. The computation of γ_{ij} is therefore dependent on the location of $(i+j)\%m$ (see Fig. 2). In other words, if $0\le(i+j)\%m\le i$, this means that ξ will be requested in the next PI and transmitted in the one after. This is due to the fact that a packet can not be transmitted in a PI by a client unless it is first reported to the server in the pervious PI. Hence, ξ will have to wait for a full PI before transmission, as well as the remaining PI before being requested. As a result, the total amount of gaps or vacations exhibited in such scenario is equal to jv_c+v_s (for the current PI) plus mv_c+v_s (the average vacation time in one PI).
Similarly, if $i+1\le(i+j)\%m\le m-1$, then ξ will be reported in the current PI, and transmitted in the one after. As a result, the total vacation time exhibited in this scenario will be equal to jv_c (for the current PI) plus mv_c+v_s.
Accordingly, we can evaluate γ_{ij} by the following:

$$\gamma_{ij} = \begin{cases}(j+m)v_c+2v_s & (0\le(i+j)\%m\le i)\\ (j+m)v_c+v_s & (i+1\le(i+j)\%m\le m-1)\end{cases}$$

Since each client has a total mean rate of λ/m, hence each arriving packet ξ can belong to any client with probability $1/m$. This implies that the conditional probability that ξ arrives at client $(i+j)\%m$ during $AI_i(t)$ is also $1/m$. Hence, $E\{Y_i(\xi)|\pi_i^\xi\} =$

$$\frac{1}{m}\begin{cases}\displaystyle\sum_{j=0}^{m-1}[(j+m)v_c+v_s] & (i=0)\\[3ex] \displaystyle\sum_{j=0}^{m-i-1}[(j+m)v_c+v_s] + \sum_{j=m-i}^{m-1}[(j+m)v_c+2v_s] & (1\le i<m)\end{cases}$$

where $\pi_i^\xi = \lim_{t\to\infty}\pi_i^\xi(t)$ (defined in Eq. (21)).
As a result, and with some simplifications, we obtain the value of $\overline{Y_i}$ as follows:

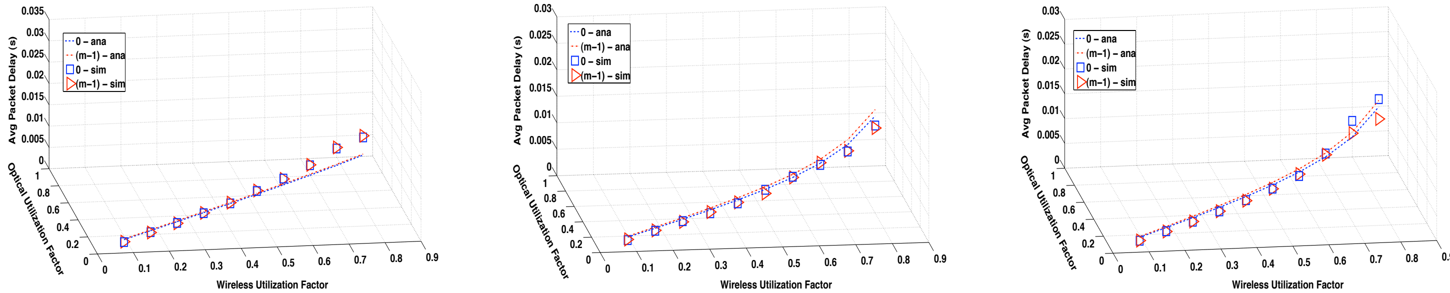
$$\overline{Y_i} = \frac{1}{2}\begin{cases}(3m-1)v_c+2v_s & (i=0)\\ (3m-1)v_c+2(1+\dfrac{i}{m})v_s & (1\le i<m)\end{cases} \quad (22)$$

Finally, by gathering the values of $\overline{R_i}$, ψ_i, and $\overline{Y_i}$ from Eqs. (20), (21) and (22) respectively, we can obtain the expected queueing delay $E\{W_{c,i}\}$ for each class $c: 1\le c\le n-1$. Using $\overline{W_{c,i}}$ in both domains, we also get $\overline{D_c}$(Fi) and $\overline{D_c}$(Wi). Note that in the wireless domain, OFDM frames have fixed sizes (5, 10 or 20 ms) [8]. Thus in the case where the OFDM uplink sub-frame is not fully utilized, each class c will exhibit an additional vacation/idle time proportional to the frame size. This time can be easily derived by measuring the uplink sub-frame utilization, to then be added to v_s. Due to space limitation, we do not show this derivation in this paper.

C. Per-flow Average Queue Size

Using Little's theorem [6], the average queue of class c size of each client i's, denoted $E\{Q_{c,i}\}$, is given by:

$$E\{Q_{c,i}\} = \overline{Q_{c,i}} = \lambda_{c,i}\overline{W_{c,i}} \quad (23)$$



(a) $c = 1$: UGS (b) $c = 2$: rtPS (c) $c = 3$: nrtPS

Fig. 4: Average End-to-End Packet Delay: $\overline{D_c}$(E2E)

TABLE I: Simulation Parameters

EPON	Number of ONU-BSs (m_o)	16
	Channel Speed (R_N)	1 $Gbps$
	Distance (OLT to ONU-BS)	25 km
	T_{up}	2 ms
	T_g	1 μs
WIMAX	Channel Bandwidth	20 MHz
	T_{ir}	0.1 ms
	$T_f = T_{up} + T_{dl}$	5 ms
	T_g	12.5 %
	Distance (SS to ONU-BS)	5 km
VPN	Number of VPNs (K)	4
	β	1
	α	0.1

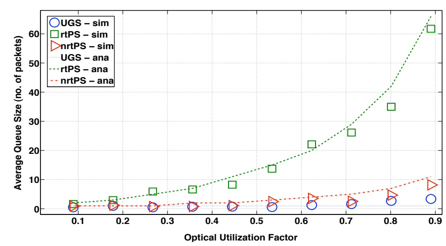


Fig. 5: Optical Average Queue Size

IV. Numerical Results

To evaluate the effectiveness of the proposed analytical model vs. the simulated proposed framework, we have implemented a simulator using OMNET++. The relevant simulation parameters are shown in Table I. Every UGS flow is generated with a mean/guaranteed rate of 64 $Kbps$ [5]. Each rtPS flow is generated at a guaranteed rate of 5 $Mbps$ (which is the average bit rate of a DVD-quality video [5]) and each nrtPS flow is generated at a guaranteed rate of 500 $Kbps$ [1]. Each self-similar pareto-shaped BE flow is generated at a mean rate of 2 Mbps [1]. Packet sizes are uniformly distributed between 64 and 1518 $bytes$. The 95% confidence interval of the simulation results gives a 7% result variation, which is statistically insignificant, and is thus not shown in the figures. Fig. 4 depicts the end-to-end (E2E) average packet delay, obtained by theoretical and simulation experiments, for 3 classes of services (CoS) of clients $i = 0$ and $i = m-1$, vs. the wireless and optical network loads. Note that in our simulation model, unlike the assumption made in the analytical model, BE traffic load builds up gradually with time. This fact results in a very slight increase of the E2E packet delay in the analytical part vs. the simulation one, at moderately lower network loads ($0.1 \rightarrow 0.3$). Nonetheless, both results in general match with a fine degree at all loads, even when the BE reserved portion is fully utilized. Moreover, although our simulation model has been implemented without the assumptions made in the analytical model, the results show that these assumptions are statistically requisite.

Furthermore, We have plotted in Fig. 5 the average queue size of each CoS in the optical domain, where each optical server acts as a multiplexer of all traffic arriving from the wireless clients connected to it. As shown, both the analytical and simulation results show a fine match of almost 95% for all classes; which again shows the effectiveness of our analysis. The wireless average queue size shows similar behavior, and is therefore not plotted.

V. Conclusion

In this paper, we have presented a new QoS-provioning framework to support layer-2 VPNs over FiWi networks. We also proposed a new generic analytical model to approximate the performance of these VPNs. Our numerical and simulation results exhibit similar outcomes, and prove the correctness of our analysis. We conclude that our model could be used to provide a fine estimation of the network behavior, to ultimately perform parameter tuning and robust protocol design.

References

[1] A. R. Dhaini, P.-H. Ho, and X. Jiang, "QoS-Aware Layer-2 VPNs over EPON-WIMAX," in *Proceedings of IEEE International Conference on Communications (ICC'10)*, Cape Town, South Africa, May 2010.

[2] G. Shen, R. S. Tucker, and C.-J. Chae, "Fixed Mobile Convergence Architectures for Broadband Access: Integration of EPON and WiMAX," *IEEE Communications Magazine*, vol. 45, no. 8, pp. 44–50, Aug. 2007.

[3] K. Yang, S. Ou, G. Ken, and H.-H. Chen, "Convergence of Ethernet PON and IEEE 802.16 Broadband Access Networks and its QoS-Aware Dynamic Bandwidth Allocation Scheme," *IEEE JSAC*, vol. 27, no. 2, pp. 101–116, Feb. 2009.

[4] W. Luo, C. Pignataro, A. Y. H. Chan, and D. Bokotey, "Layer-2 VPN Architecture," *CISCO Press*, 2005.

[5] A. R. Dhaini, C. M. Assi, M. Maier, and A. Shami, "Per-Stream QoS and Admission Control in Ethernet Passive Optical Networks (EPONs)," *IEEE/OSA Journal of Lightwave Technology (JLT)*, vol. 25, no. 7, pp. 1659–1669, July 2007.

[6] D. Bertsekas and R. Gallager, *Data Networks*. Prentice-Hall, 1992.

[7] X. Bai, A. Shami, and Y. Ye, "Delay Analysis of Ethernet Passive Optical Networks with Quasi-leaved Polling and Gated Service Scheme," in *Proceedings of AccessNets'07*, Ottawa, ON, Canada, Aug. 2007.

[8] "IEEE Standard for Local and Metropolitan Area Networks Part 16: Air Inteface for Fixed Broadband Wireless Access Systems," Online, 2004.