Delay aware load balancing over multipath wireless networks

Oscar Delgado, *Member, IEEE,* and Fabrice Labeau, *Senior Member, IEEE,*

***Abstract*—The ability of mobile devices to be connected to more than one radio node at the same time enables mobile devices to transmit & receive traffic to & from multiple paths. This ability helps to increase the average mobile device data rate and to improve the network reliability. Load balancing among multiple paths become a key factor to avoid network congestion, nevertheless it requires efficient techniques to split traffic without adding more delay or generating too much packet reordering for delay sensitive traffic.**

In this paper, we address two key issues in the context of uplink wireless mobile networks: 1) How to accurately split traffic among multiple paths and 2) how to minimize the end-to-end delay without increasing packet reordering. We propose delay aware load balancing algorithm (DALBA), a novel strategy that splits traffic at the granularity of packet. DALBA aims to minimize the splitting error and the end-to-end delay difference by effectively using all the available paths. We analyze DALBA's performance through extensive simulations using H.264 video traffic. Numerical results demonstrate that DALBA outperforms previous algorithms in terms of splitting error, end-to-end delay and peak signal-to-noise ratio while keeping packet reordering to a suitable low value.

***Index Terms*—DALBA, load balancing, packet reordering, end-to-end delay, multi-path, uplink, 5G.**

I. INTRODUCTION

Recent advances in technology have opened the possibility to establish multiple radio access paths between end devices. Some of the advantages of using multiple paths include increasing network capacity, guaranteeing high quality-of-service (QoS) and avoiding single points of failure [1], [2] (e.g. bandwidth, delay and reliability) imposed by real-time multimedia applications.

The rapidly growing demand of real-time multimedia applications over mobile wireless networks, e.g. live streaming video, video conference, multi-player on-line gaming, etc., opens the possibility to consider multi-path transport as a promising solution due to its many benefits, including high throughput and improved reliability [3].

The main research challenge in utilizing multiple paths in the context of uplink wireless networks is to accurately split input traffic so as to provide acceptable QoS perceived by end users. Inefficient load balancing can significantly degrade the network performance thus creating large end-to-end delay, packet reordering, etc.

The purpose of load balancing is to optimize resource use, i.e. minimize end-to-end delay and maximize throughput. Using multiple paths to the destination with load balancing instead of a single path may help to increase availability and reliability through redundancy. The ability to use multiple paths simultaneously increases the available bandwidth [4].

Packet reordering arises when the packets that arrive at the destination have different order as compared to the same packets at the source [5], [6]. Fig. 1 shows an example of a situation in which packets 2, 3 and 4 arrived at the destination out-of-order.

Packet reordering affects both transmission control protocol (TCP) and user datagram protocol (UDP) based applications [7]. For TCP-based applications, duplicate acknowledgements are sent when out-of-order packets are received, causing retransmission of packets and a decrease in the effective transmission rate. For UDP-based applications, out-of-order packets that arrive at the destination after the playout deadline are discarded [8].

Although it is known that packet reordering might degrade the quality of service in IP networks, its extent is still not completely understood. An early effort to quantify the effects of packet reordering on IPTV using MPEG-2 video codec revealed that the video quality becomes unacceptable for more than 0.12% of out-of-order packets [9].

The end-to-end delay is the time it takes to deliver a packet from the source to the destination. To understand the negative effects of end-to-end delay on multimedia quality, we take for example the case of a video conference transmission. Timing is an important characteristic of video. Two frames of a video are separated with an interval. This interval is as much a part of the video as the frame itself. If additional delay is inserted between frames, the rhythm of the video is lost. Long end-to-end delay may force the video conference to freeze frames degrading the quality of the conference. Therefore end-to-end delay difference among paths plays an important role in the overall quality of any multimedia application [10].

This paper proposes delay aware load balancing algorithm (DALBA), a novel solution for multi-path uplink heterogeneous wireless mobile systems that splits traffic at a packet granularity. We formulate the load balancing problem as a constrained multi-objective optimization problem and propose a solution based on the multi-level programming method. DALBA is specifically tested in the context of UDP H.264 video streaming (real-time and non real-time). Simulation

O. Delgado is with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC, H3A-0G4 CANADA e-mail: oscar.delgadocollao@mail.mcgill.ca

F. Labeau is with the Department of Electrical and Computer Engineering, McGill University, Montreal, QC, H3A-0G4 CANADA e-mail: fabrice.labeau@mcgill.ca

Manuscript received Month xx, 2015; revised Month yy, 2016.

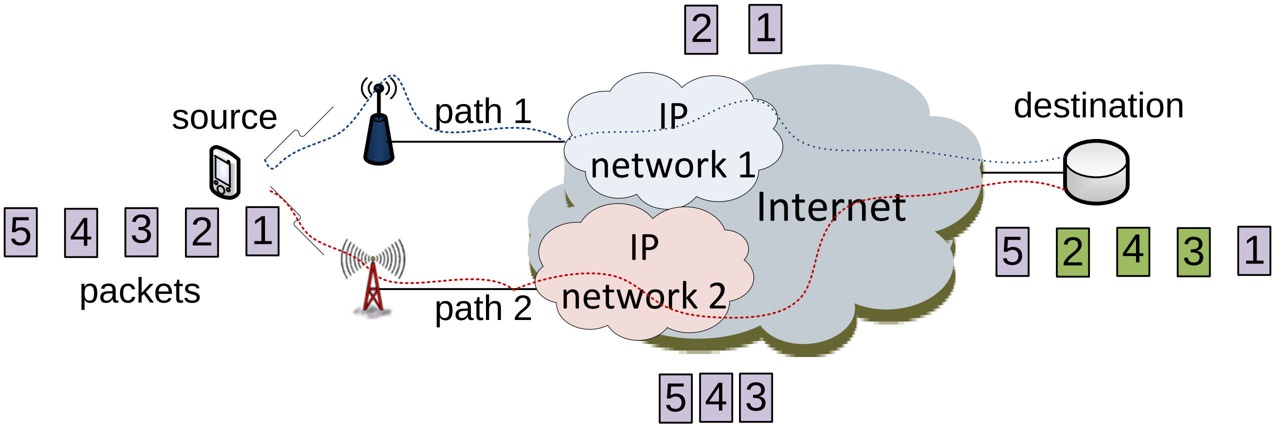

Fig. 1. Multi-path scenario: Mobile device attached to two different networks experiencing packet reordering.

results demonstrate that DALBA:

- Accurately splits traffic while reducing the average packet loss and the average end-to-end delay.
- Is robust to different traffic loads. It exhibits superior PSNR performance in high traffic load conditions.
- Outperforms previous algorithms in total end-to-end delay and reduces packet reordering.

The paper is organized as follows. In Section II, we briefly summarize previous works on load balancing. In section III we present a multi-objective optimization model of the load balancing problem for heterogeneous wireless uplink systems. In Section IV, we propose delay aware load balancing algorithm (DALBA). In Section V we present and analize the simulation results. Final conclusions are presented in the last section.

II. RELATED WORK

Traffic load balancing over multi-path networks has been an active research area in recent years [11]–[13]. The existing models can be loosely categorized into packet-based and flow-based load balancing models.

A. Flow-based load balancing

The flow-based load balancing models address the problem by assigning packets of the same flow to the same path. Although the risk of packet reordering decreases, queuing packets over the same path causes the end-to-end delay to increase.

Flowlet Aware Routing Engine (FLARE) [14] proposed to group packets into small subsets of packets called flowlets, and to use these flowlets as the scheduling unit. Flare defines a threshold time that helps to avoid packet reordering. One drawback of FLARE is that it does not take into account packet loss. Packet loss is especially important when transmitting TCP traffic because FLARE may retransmit packets in saturated paths.

Another approach was presented in [15]. In [15] the authors investigate the challenge of splitting a traffic flow over the worldwide interoperability for microwave access (WiMAX) and the local area wireless computer networking technology (Wi-Fi) links and proposed an airtime-balance method. This method maps the traffic load to an airtime cost function and uses it to split traffic. The idea is to send packets to radio nodes so that the airtime cost is balanced. However, this solution does not deal with packet reordering.

Adaptive Load Balancing Algorithm (ALBAM) [16] focused on the TCP drawbacks found in FLARE and proposed an algorithm that schedules traffic only when the packet inter-arrival time is able to compensate for the path delay difference. Despite de fact that accurate delay estimations must be available, ALBAN does not evaluate the overall end-to-end delay.

Flow Slice (FS) [17] proposed to subdivide every flow into smaller pieces (flow slice) at every inter-flow slice interval larger than a predefined threshold and to balance the traffic load on a flow slice granularity. FS reduces the probability of packet reordering at the cost of high end-to-end delay.

QBALAN [18] proposed an algorithm that subdivides the task of load balancing in two parts: First it schedules part of the traffic that remains on the system for long periods of time (long-term strategy), and then it takes the rest of the traffic and schedules it, so that the overall splitting error is minimized (short-term strategy). Although QBALAN manages to effectively reduce the average splitting error and the percentage of packet reordering, its end-to-end delay performance suffers when it experiences high levels of traffic load.

B. Packet-based load balancing

The packet-based load balancing models manage to reduce the overall end-to-end delay, by using packets as the basic allocation unit. This strategy reduces their ability to maintain low levels of packet reordering.

Effective Delay Controlled Load Distribution (E-DCLD) [19] realized that inefficient load balancing can degrade the network performance and tackles the problem by formulating an optimization problem that balances the end-to-end delay among all the available paths. One of the main concerns of E-DCLD is the low convergence time.

Convex optimization-Based Method (CBM) [20] handles the low convergence time experienced by E-DCLD by formulating the load balancing problem as a convex optimization problem.

This is the author's version of an article that has been published in this journal. Changes were made to this version by the publisher prior to publication.
The final version of record is available at http://dx.doi.org/10.1109/TVT.2017.2655011

Although CBM solves the low convergence time issue, its main disadvantage is that the input traffic is modeled as a Poisson distribution but this assumption is not suitable for most video traffic applications.

An interesting approach is studied in [21]. In [21] the authors proposed a machine learning algorithm that calculates the load balancing splitting ratio from the available QoS information, i.e., delay, throughput, buffer size, packet loss, etc. One important disadvantage of the machine learning algorithm that we can mention is that although it estimates the optimal load balancing ratio from the available QoS information, the algorithm may not quickly find the new optimal load balancing ratio. For example if some of the paths experience sudden congestion and QoS information quickly changes.

Sub-Packet based Multipath Load Distribution (SPMLD) [22] proposed to formulate the problem as a constrained optimization problem that minimizes the end-to-end delay. Its main idea is to reduce packet reordering by grouping multiple paths into a single virtual path. One of the main concerns of SPMLD is its complexity, since it proposes two distributed algorithms that have to be implemented in the source and the destination nodes.

To summarize, flow-based models manage to limit packet reordering to a negligible level, but at the cost of large end-to-end delay. On the other hand, packet-based models manage to reduce end-to-end delay, but their ability to reduce packet reordering has to be addressed especially when the traffic load is high.

III. PROBLEM STATEMENT

In this section, we formulate a multi-objective optimization model for the load balancing problem over multiple paths. We consider the uplink case of a wireless network in which mobile devices have more than one network interface and can be attached to multiple radio nodes at the same time, i.e., traffic generated in a mobile device can be routed to the destination through multiple radio nodes. We assume that each radio node knows the channel capacity; that is the maximum rate at which a mobile device can send traffic. Channel capacity information is reported to every mobile device by the radio nodes.

Let $\mathcal{N}$ be the set of radio nodes, $\mathcal{M}$ the set of mobile devices and $\mathcal{W}$ the set of flows. In a given TTI (transmission time interval), any flow $w \in \mathcal{W}$ can be scheduled through multiple radio nodes $n \in \mathcal{N}$. Path n refers to the connection between mobile device m and the destination passing through radio node n. The main objective is to find the appropiate packet assignment such that we sent traffic to radio nodes minimizing the splitting error and the end-to-end delay while limiting the negative effects of packet reordering. Each mobile device m is attached‘ to multiple radio nodes, see Fig. 2.

At a given point in time, let $\mathcal{W}_m$ be the set of flows in mobile device m, $\mathcal{N}_m$ the set of radio nodes to which mobile device m is attached to. $\mathcal{P}_w$ be the set of packets of flow w in mobile m's buffer ready to be scheduled, $\alpha_{p_w,w,n} \in \{0,1\}$ be the packet assignment indicator, $\alpha_{p_w,w,n} = 1$ if packet $p_w \in \mathcal{P}_w$ is assigned to radio node n, G_n^m be the channel capacity, and $\hat{R}_n^{m^t}$ be the rate estimate of the traffic sent from mobile device m to radio node n at time t, $\hat{R}_n^{m^t}$ is constantly updated on each mobile device m by an the exponential moving average [23] with parameter $\gamma \in (0,1)$:

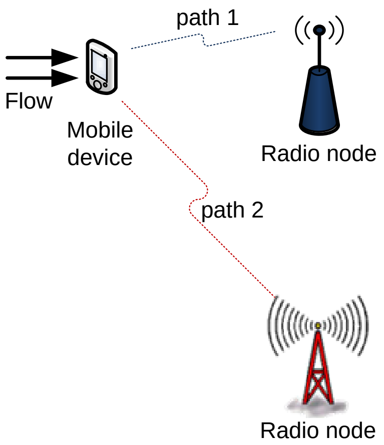

Fig. 2. Basic scenario: Mobile device attached to two radio nodes, transmitting two traffic flows.

$$\hat{R}_n^{m^t} = \gamma \sum_{p_w \in \mathcal{P}_w, w \in \mathcal{W}_m} \alpha_{p_w,w,n} R^t_{p_w,w} + (1-\gamma)\hat{R}_n^{m^{t-1}} \qquad \forall n \in \mathcal{N}_m, \tag{1}$$

where $\hat{R}_n^{m^{t-1}}$ is the rate estimate in the previous TTI and $R^t_{p_w,w}$ is the instantaneous packet rate. An exponential moving average $\hat{R}_n^{m^t}$ as defined in equation (1) is used to quickly estimate longer-term trends. Notice that this definition of $\hat{R}_n^{m^t}$ is used to smooth the estimated rate, i.e., it is bias towards long term trends in order to avoid being affected by short term fluctuations. For simplicity, we suppress the explicit dependence on time t in the notation. Given a set of available paths $n \in \mathcal{N}_m$ we define the splitting ratio as:

$$\Psi_n^m = \frac{G_n^m - \hat{R}_n^m}{G_n^m}. \tag{2}$$

The end-to-end delay $D_{n,w}^m$ of flow w can be expressed as $D_{n,w}^m = f(\alpha_{p_w,w,n})$, making explicit that the end-to-end delay depends on the assignment indicator $\alpha_{p_w,w,n}$, and that in order to reduce packet reordering the end-to-end delay difference should be as small as possible.

The objective is to find the value of the packet assignment indicator $\alpha_{p_w,w,n}$ that minimizes the splitting ratio Ψ_n^m and the end-to-end delay $D_{n,w}^m$ at the same time. Therefore the optimal packet assignment indicator $\alpha_{p_w,w,n}$ can be found by solving the following multi-objective optimization problem:

$$\min_{\alpha_{p_w,w,n}: w \in \mathcal{W}_m} \left\{ \max_{n \in \mathcal{N}_m} \left\{ \left(\Psi_n^m, D_{n,w}^m \right) \right\} \right\} \qquad \forall m \in \mathcal{M} \tag{3}$$

subject to:

$$\hat{R}_n^m \leq G_n^m \qquad \forall n \in \mathcal{N}_m \tag{4}$$

$$\sum_{n \in \mathcal{N}_m} \alpha_{p_w,w,n} = 1 \qquad \forall p_w \in \mathcal{P}_w, \forall w \in \mathcal{W}_m \tag{5}$$

$$\begin{aligned} &\alpha_{p_w,w,n} \in \{0,1\} \\ &\qquad \forall p_w \in \mathcal{P}_w \forall w \in \mathcal{W}_m, \forall n \in \mathcal{N}_m. \end{aligned} \tag{6}$$

Notice in the min-max equation (3) that $(\Psi_n^m, D_{n,w}^m)$ is a vector of objectives of the form $F(x) = (f_1(x), f_2(x))$. Constraint (4) ensures that packets are only assigned to a specific path if the corresponding rate $\hat{R}_n^m$ does not exceed the maximum rate G_n^m. Constraints (5) and (6) refers to the fact that a given packet can only be assigned to one path at a time.

In order for the set of equations (3) - (6) to have a feasible solution, there should exist at least one path n that satisfies equation (4). If there is no practical solution, packets of flow $w \in \mathcal{W}_m$ will be dropped. Radio node n sends G_n^m and $D_{n,w}^m$ information to all mobile devices m every δms.

Solving a multi-objective optimization problem is a challenging task. Nevertheless, there exist multiple methods to deal with multi-objective optimizations [24]. A classical approach is to formulate the multi-objective optimization problem as a single-objective optimization problem by means of scalarization such that the optimal solutions are Pareto optimal. A common difficulty with this method is to find the most appropriate parameters for the scalarization (different parameters means different optimal solutions).

Our proposed solution is related to the multi-level programming method. Multi-level and especially bi-level optimization method is designed to deal with problems with two objectives in which the optimal decision of one of them is constrained by the decision of the second objective [25]. The second-level objective optimizes its solution under a feasible region that is defined by the first-level objective. In our problem the bi-level optimization method can be roughly modelled as in equations (7)-(8): minimize the end-to-end delay $D_{n,w}^m$ (second-level objective) subject to: maximize the splitting ratio Ψ_n^m (first-level objective) [26], [27].

$$\min \ D_{n,w}^m \tag{7}$$

$$\text{s.t. } \max \ \ \Psi_n^m \tag{8}$$

When solving this multi-objective optimization problem we should remember that in general, specially in the context of real-time applications, the problem of finding the optimal solution to the load balancing problem is NP-hard [28], [29], therefore intractable when the number of packets exceeds a few units. This is the reason why we focus our attention to the use of heuristics in the next section.

IV. DELAY AWARE LOAD BALANCING ALGORITHM (DALBA)

In order to solve optimization problem (3) - (6) we propose the load balancing algorithm DALBA implemented at the source mobile device. DALBA can be described in two main steps loosely corresponding to the bi-level optimization method:

- *First:* Assign the portion of traffic of flow $w \in \mathcal{W}_m$ to each path $n \in \mathcal{N}_m$ so as to minimize the splitting ratio Ψ_n^m difference among paths.
- *Second:* Assign packets of flow w to each path n so as to minimize the end-to-end delay $D_{n,w}^m$.

In order to appropriately choose the path to be used by every packet, we calculate the ratio λ_n^m as the product of the splitting ratio Ψ_n^m and the actual load G_n^m.

$$\lambda_n^m = \frac{\Psi_n^m \cdot G_n^m}{\sum_j (\Psi_j^m \cdot G_j^m)}. \tag{9}$$

Replacing equation (2) in equation (9), λ_n^m can be presented as:

$$\lambda_n^m = \frac{G_n^m - \hat{R}_n^m}{\sum_j (G_j^m - \hat{R}_j^m)}. \tag{10}$$

Ratio λ_n^m is the normalized difference between the maximum rate at which mobile device m can send traffic and its actual load. The intuition behind the ratio λ_n^m is based on the realization that in order to balance traffic load we should transmit traffic on each path proportionally to the available capacity $G_n^m - \hat{R}_n^m$.

The proposed algorithm DALBA runs in every mobile device. It works as follows: At every TTI, for every flow, it calculates the splitting ratio λ_n^m, then it evaluates if there exist enough available capacity among all the available paths to receive all the packets of the flow (if not, packets of that flow are discarded). if there are enough capacity it assigns packets to paths according to the splitting ratio (first main step).

Having the path ratio decided, DALBA decides which packets should be sent first (remember that it is important the order in which packets are sent to avoid packet reordering). While there are packets to be sent. DALBA selects the path that has the smallest end-to-end delay and sends the first packet in queue on that path (second main step). Notice that the end-to-end $D_{n^*,w}^m$ must be updated at every iteration inside the while loop by considering that a packet has being assigned for scheduling on path n^* (scheduling delay). The load balancing algorithm is described in Algorithm 1.

A. Complexity Analysis

It is worth noticing that flows are composed of packets. Therefore, packets are the smallest balancing units used by algorithm DALBA. The proposed algorithm allocates each packet after performing a linear search on the path that minimizes the end-to-end delay. Hence, the complexity to allocate the first packet is $\mathcal{O}(N)$, the complexity to allocate the second packet is $\mathcal{O}(N)$, and so on until all packets are allocated. Consequently, the total complexity of the algorithm is $\mathcal{O}(NP)$. i.e. the algorithm has linear complexity in the number of paths and in the number of packets, and thus could be easily implemented in real-time.

V. SIMULATION SCENARIO AND RESULTS

In this section, we present the simulation settings used to assess the performance of our load balancing algorithm DALBA and then we analyze the experimental results.

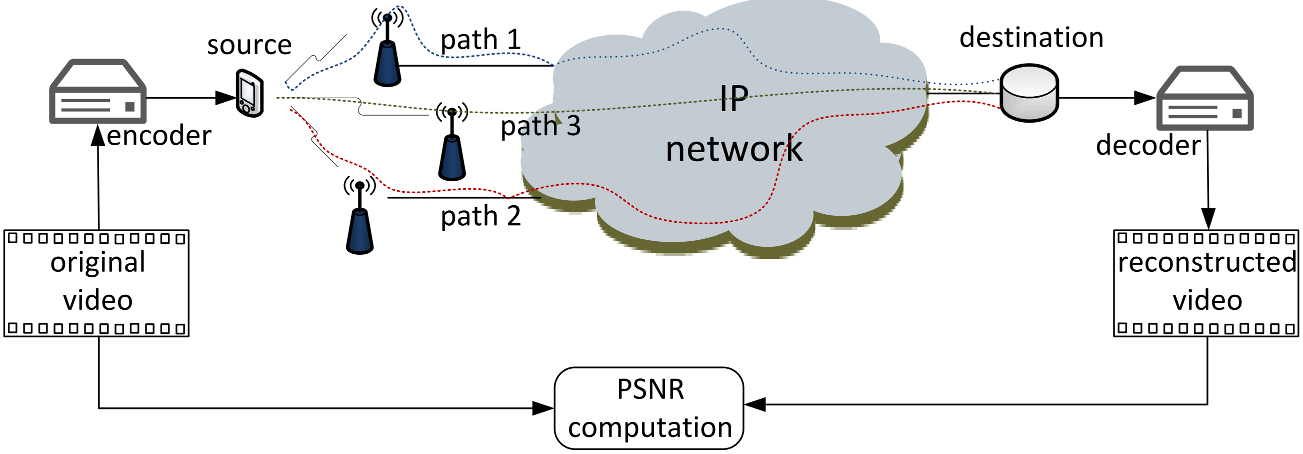

Fig. 3. Simulation setup: basic architecture for simulations.

Algorithm 1 DALBA

```
Every TTI
$\alpha_{p_w,w,n} \leftarrow 0, \forall p_w \in \mathcal{P}_w, \forall w \in \mathcal{W}_m, \forall n \in \mathcal{N}_m$
for every flow $w \in \mathcal{W}_m$ do
  Calculate the ratio $\lambda_n^m$ according to (10) $\forall n \in \mathcal{N}_m$
  if $\exists$ $n$ such that $\hat{R}_n^m < G_n^m$ then
    $sPacket_n \leftarrow |\mathcal{P}_w| \times \lambda_n^m, \forall n \in \mathcal{N}_m$
    $counter_n \leftarrow 0, \forall n \in \mathcal{N}_m$
    while $\mathcal{P}_w \neq \emptyset$ do
      Find path $n^* = \arg\min_n \{D_{n,w}^m\}$
      if $counter_{n^*} < sPacket_{n^*}$ then
        $\alpha_{p_w,w,n^*} \leftarrow 1$ (assign packet $p_w \in \mathcal{P}_w$ to radio node $n^*$)
        $counter_{n^*} \leftarrow counter_{n^*} + 1$
        $D_{n^*,w}^m \leftarrow$ Update (scheduling delay)
        $\mathcal{P}_w \leftarrow \mathcal{P}_w \setminus \{p_w\}$
      else
        $D_{n^*,w}^m \leftarrow \infty$
      end if
    end while
  else
    Discard packets of flow $w$
  end if
end for
```

A. Evaluation environment

We evaluate DALBA using trace-driven simulations in MATLAB. Our data set contains videos comprised of action, comedy and drama movie trailers. The architecture for simulations is depicted in Fig. 3.

1) Simulation setup: We consider an LTE wireless mobile cellular system with 30 radio nodes and 100 mobile devices in which each mobile device is simultaneously connected to multiple radio nodes. We have two scenarios. In scenario 1, all mobile device's are connected to 3 radio nodes, and in scenario 2, 50% of the mobile device's are connected to 2 radio nodes and 50% are connected to 3 radio nodes. We show the general simulation network topology in Fig. 4.

We assume that all radio nodes belong to the same network operator, and that radio nodes have a perfect knowledge of the maximum rates (channel capacity) a mobile device can transmit on each path. These information is provided to each mobile device by the radio nodes at least every δms. We also assume that mobile device's are able to estimate the end-to-end delay at least every δms.

Our experiments assume on demand streaming H.264 videos. Ffmpeg software is adopted as the video codec tool [30]. The movie trailers are encoded with variable bit rate (VBR) using mp4 format at 24 fps (frames per second). To ensure diversity on the movie trailers used we uniformly choose them among 3 video categories (action, comedy and drama). A GOP (group of pictures) consists of 12 frames (IBBPBBPBBPBB). Input traffic on each mobile device is composed of four video streams. The video frames are transmitted to the destination using UDP packets. We further assume that mobile devices are static and that the buffer size at the destination is large enough to accommodate all the arriving packets. Hence, packet loss could only be caused by late packets.

In our simulations we use the concepts of decoding threshold time and traffic load. The decoding threshold time is defined as the maximum allowable time a packet can remain in the buffer after the playout deadline. We use a decoding threshold time of 500ms for a video on demand service that buffers 500ms and then plays out the video. Traffic load is the total traffic carried by the network and it is represented as a percentage of the total network capacity.

In order to model the total multi-hop end-to-end delay of any given path between a mobile device and the destination we use the Pareto distribution. The Pareto distribution offers a close approximation to the real end-to-end delay with low complexity [31], [32]. Table I summarizes the simulation parameters.

2) Compared load balancing schemes: There are a number of proposed solutions for the load balancing problem. In this paper we compare DALBA with the following three existing load balancing models:

- **Flowlet Aware Routing Engine (FLARE)** [14] groups packets into flowlets and uses these flowlets as the

TABLE I
SIMULATION PARAMETERS

Parameter	value
Simulation time	100s
Radio nodes	30
Mobile devices	100
Mobile device location	uniform distribution
Mobile device speed	static
Paths per mobile device (scenario 1)	3
Paths per mobile device (scenario 2)	2 (50%), 3 (50%)
Flows per mobile device	4
Traffic type	UDP
Delay difference between paths	5ms
Delay variance	1ms^2
Average video rate per flow	1.6Mbps
Frames per second	24
Group of pictures (GOP)	12 frames (IBBP..BBP)
Video category	action, comedy, drama
δ	500ms
γ	0.001

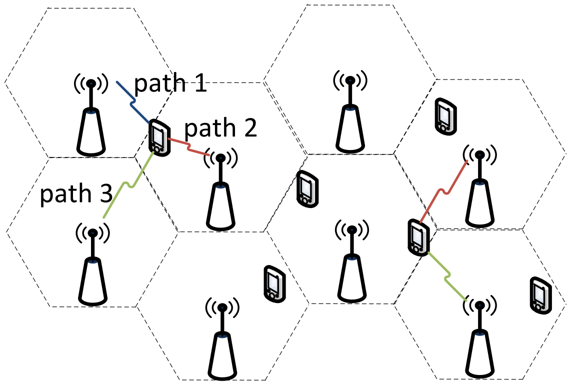

Fig. 4. Simulation network topology: 30 radio nodes, 100 mobile devices uniformly distributed.

balancing unit. The round trip delay is checked every 500ms.

- **Sub-Packet based Multipath Load Distribution (SPMLD)** [22] minimizes the end-to-end delay solving a constrained optimization problem. The predominant parameters C_1, C_2 and C_3 are set to 1.2, -0.17 and -0.005 respectively.
- **Load balancing algorithm (QBALAN)** [18] subdivides flows into two groups, short-term and long-term, and uses these groups to minimize the splitting error and packet reordering. The round trip delay is checked every 500ms for the short-term load balancing algorithm, and every 7000ms for the long-term load balancing algorithm.

3) Evaluation metrics:

- **Splitting error (SE)** measures the fluctuation of the desired load with respect to the actual load in each path. The smaller the value is, the more accurate load balancing we have. We can calculate the splitting error SE as:

$$SE = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \frac{1}{|\mathcal{N}_m|} \sum_{n \in \mathcal{N}} \frac{|K_n^m - \hat{R}_n^m|}{K_n^m}, \tag{11}$$

where the desired load K_n^m is a system parameter that represents the amount of traffic expected to be transmitted by mobile m on path n such that it keeps the load at the desired level and it is assumed to be known, e.g., if three radio nodes are expected to receive the same amount of traffic from mobile m then $K_n^m = \frac{1}{3}\sum_n \hat{R}_n^m$.
- **End-to-end delay** (D) measures the time taken for a packet to be transmitted across a network from source to destination. It includes the transmission delay D_{tx}, the propagation delay D_p, the processing delay D_{pc} and the queuing delay D_q.

$$D = D_{tx} + D_p + D_{pc} + D_q. \tag{12}$$

- **Reorder Density (RD)** [33], this metric measures the amount of packet reordering in the arriving packet sequence. RD metric is defined as the distribution of packet displacements. Negative displacements in the RD distribution represent the number of positions by which the received packet is early, while positive displacements represent the lateness of the received packets.
- **Mean displacement of packets** (M_D) [34]. When calculating the mean of the reordering density RD, if all packets are included, the mean becomes zero. Therefore, the mean, when all packets are taken together, is not useful. Instead we can consider the magnitude to define a mean displacement M_D:

$$M_D = \sum_{i=-DT}^{+DT} |i| RD(i). \tag{13}$$

- **Reorder entropy** (E_R) [34]. Entropy is a concept that is used to define the randomness or the disorder and can be used as a convenient metric for a distribution's tendency to be concentrated or dispersed. As RD is a discrete probability distribution, that of packet disorder, we can define reorder entropy as:

$$E_R = -\sum_{i=-DT}^{+DT} RD(i) \ln RD(i), \tag{14}$$

where the displacement threshold (DT) is a threshold on the displacement of packets that allows the metric to classify a packet as lost or duplicate. If there is no packet reordering, i.e., $RD(0) = 1$, the reorder entropy is equal to zero. On the other hand, if the packet sequence has the highest variance, packets are displaced uniformly with equal probabilities. Then the upper bound for the reorder entropy is $\ln(2DT + 1)$.
- **Peak signal-to-noise ratio (PSNR)** is a metric for video quality that can be defined as the ratio between the maximum possible value (power) of the original video signal and the power of the distorting noise that affects the quality of its representation at the receiver. This metric is measured using Evalvid open source tool [35].

B. Simulation results

1) Splitting error: First, we show the percentage of splitting error as a function of the traffic load for scenarios 1 and 2 in Fig. 5. The results indicate that DALBA accurately splits traffic, managing to achieve the lowest splitting error in both scenarios. Contrary to SPMLD, as the traffic load increases DALBA manages to maintain the splitting error almost constant. This is because DALBA quickly adjusts the splitting ratio such as to sent packets to the paths with smaller load.

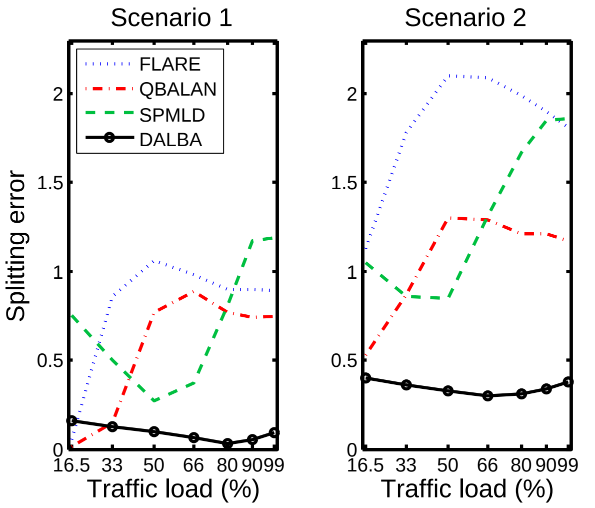

Fig. 5. Average splitting error (SE) over Traffic load for scenarios 1 and 2.

2) Packet reordering: Fig. 6 shows the reordering density at 80% of the traffic load for scenario 1. Here DALBA has lower packet reordering than SPMLD. DALBA keeps the packet reordering below five packets which is the price to pay if we want to reduce packet loss. Contrary to DALBA, QBALAN and FLARE have lower packet reordering at the cost of higher packet loss as we can observe in Fig. 9.

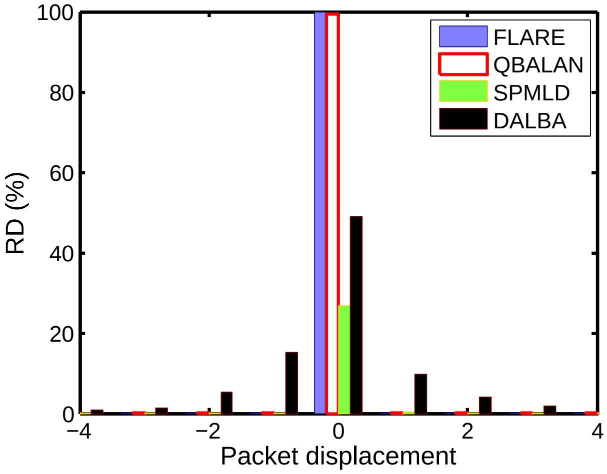

Fig. 6. Reorder density at 80% traffic load for scenario 1.

In order to have a sense of the level of dispersion of packet reordering we use the reordering entropy. Fig. 7 summarizes the reorder entropy as a function of the traffic load for both scenarios. According to Fig. 7 DALBA has lower reordering than SPMLD but higher than QBALAN and FLARE. This could be explained by the fact that DALBA tries to distribute packets of the same flow over all the available paths, which might cause packet reordering. Nevertheless, we also have to analyse how deep the reordering is, in order to asses the real packet reordering impact.

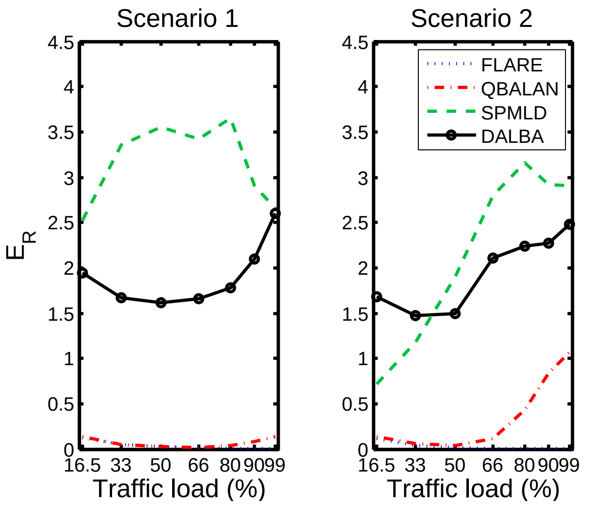

Fig. 7. Reorder entropy E_R over Traffic load for scenarios 1 and 2.

Another metric that allows us to analyze the packet reordering behaviour is the the mean displacement of packets shown in Fig. 8 for scenario 1. In this figure DALBA, QBALAN and FLARE have the lowest levels, DALBA handles to keep mean displacement of packets below 5 packets even at high traffic loads. We should remember that these metrics capture reordering at the packet level and it is not directly associated with the decoding threshold time (maximum allowable time a packet can remain in the buffer after the playout deadline). Therefore, in order to determine the acceptable level of reordering we must also verify its impact at the application level.

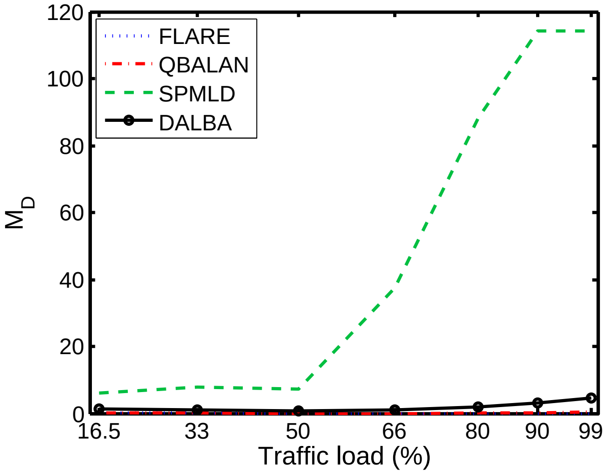

Fig. 8. Mean displacement of packets over Traffic load for scenarios 1.

Fig. 9 shows that DALBA has the smallest packet loss especially when the traffic load increases. This is due to the fact that DALBA's strategy is to deliver packets as fast as it can, therefore meeting the decoding threshold time.

3) End-to-end delay: Fig. 10 shows that DALBA has superior end-to-end delay performance as compared to the other algorithms. This can be attributed to the ability of DALBA

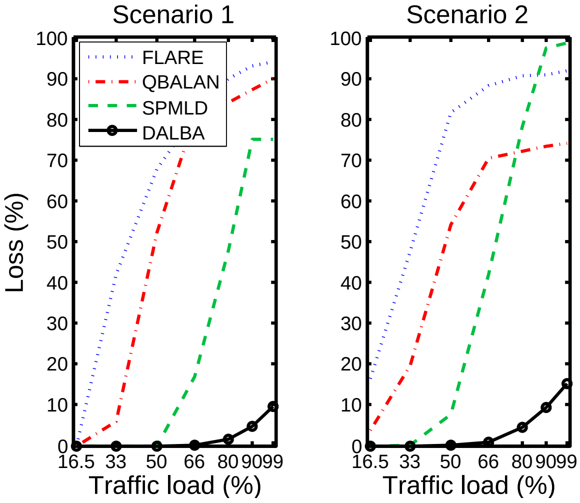

Fig. 9. Packet loss over Traffic load, non real-time (decoding threshold time 500ms) for scenarios 1 and 2.

to send packets to the paths that has smaller delay. QBALAN and FLARE manage to have smaller packet reordering due to its strategy of sending packets of the same flow by the same path, which in turn increases the end-to-end delay. SPMLD makes an effort to keep low delay but it struggles when the traffic load increases.

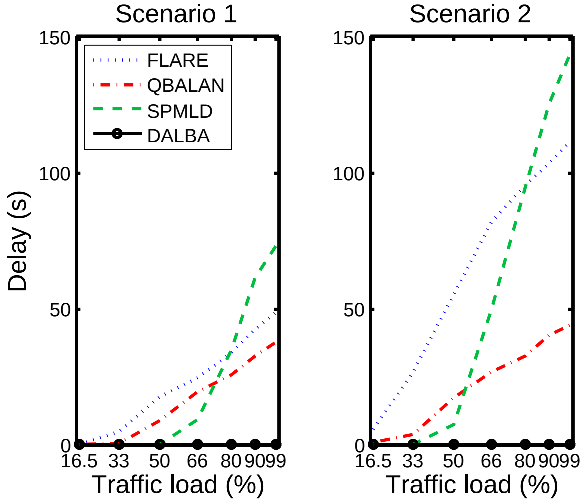

Fig. 10. End-to-end delay over Traffic load for scenarios 1 and 2.

Fig. 11 shows the end-to-end delay for scenario 1 for the packets that meet the decoding threshold time of 500ms (packets that arrive late are discarded). In this figure we can see that DALBA achieves smaller end-to-end delay. When the traffic load increases SPMLD has smaller end-to-end delay at the cost of higher packet loss, as can be seen in Fig. 9. FLARE and QBALAN reduce packet reordering by assigning packets to the same path but this strategy generates congestion and therefore higher end-to-end delay and packet loss.

4) PSNR: Fig. 12 shows the average PSNR values over different traffic loads for scenario 1. It can be noticed that DALBA has better PSNR performance. We can also notice that the PSNR pattern is almost the opposite to that of packet loss (see Fig. 9). Larger end-to-end delay leads to more packet loss and that degrades the quality of the video. It is worth mentioning that DALBA's good PSNR performance holds true for larger values of delay variance and delay differences between paths.

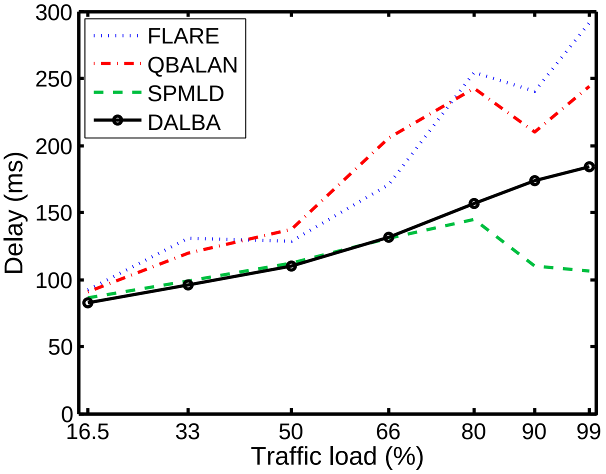

Fig. 11. End-to-end delay over Traffic load, non real-time (decoding threshold time 500ms) for scenario 1.

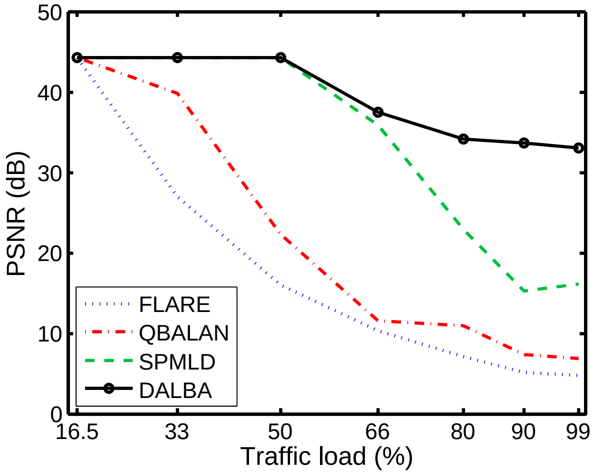

Fig. 12. PSNR over Traffic load, non real-time (decoding threshold time 500ms) for scenario 1.

DALBA outperforms the other algorithms by delivering more video frames within the decoding threshold time. We can see this especially when the traffic load increases forcing the system to efficiently use all the available paths. We also plot the PSNR distribution for scenario 1 for the case of a decoding threshold time of 500ms at 80% traffic load in Fig. 13 and the PSNR standard deviation as a function of the traffic load in Fig. 14. In both figures we can observe that DALBA achieves higher PSNR values (Fig. 13) with smaller variations (Fig. 14) while the other compared algorithms struggle to maintain its performance at high traffic loads.

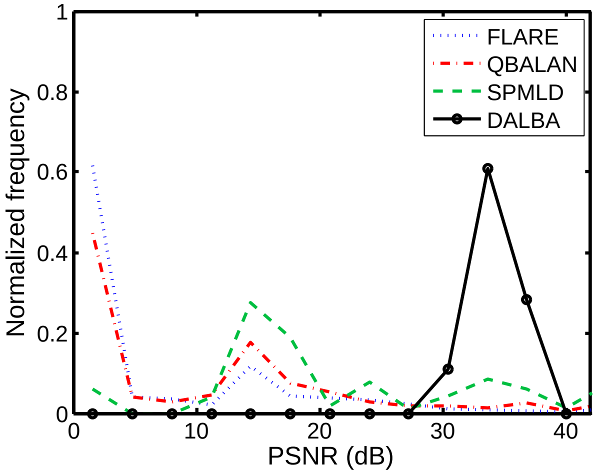

Fig. 13. PSNR distribution at 80% traffic load (decoding threshold time 500ms) for scenario 1.

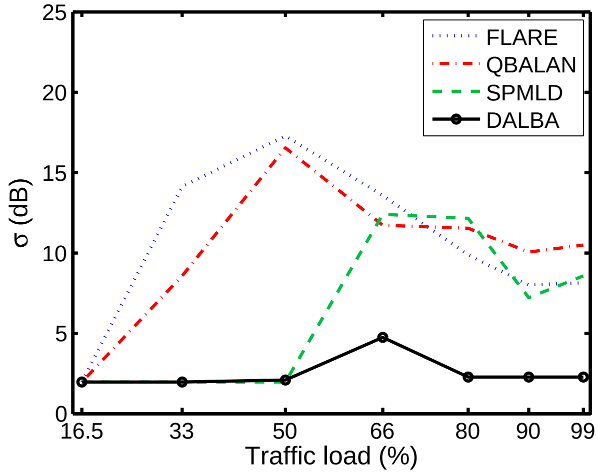

Fig. 14. PSNR standard deviation over Traffic load (decoding threshold time 500ms) for scenario 1.

VI. CONCLUSIONS

In this paper, we propose DALBA a multi-path load balancing method for heterogeneous wireless uplink systems that deals with the problem of how to distribute traffic without causing excessive packet reordering, while keeping the splitting error and the end-to-end delay as small as possible. DALBA is a sub-optimal heuristic solution specifically tested in the context of UDP multimedia applications.

In order to analyze DALBA's performance, we conduct extensive simulations in MATLAB using H.264 video streaming. Simulation results demonstrate that: (1) DALBA accurately splits traffic while reducing the average packet loss and the average end-to-end delay; (2) DALBA is robust to different traffic loads. It exhibits superior PSNR performance in high traffic load conditions; (3) DALBA outperforms previous algorithms in total end-to-end delay and reduces packet reordering.

REFERENCES

[1] N. Freris, C.-H. Hsu, J. Singh, and X. Zhu, "Distortion-Aware Scalable Video Streaming to Multinetwork Clients," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 469–481, April 2013.

[2] M. Menth, R. Martin, A. M. C. A. Koster, and S. Orlowski, "Overview of resilience mechanisms based on multipath structures," *International Workshop on Design and Reliable Communication Networks*, pp. 1–9, October 2007.

[3] T. Ernst, N. Montavont, R. Wakikawa, C. Ng, and K. Kuladinithi, "Motivations and Scenarios for Using Multiple Interfaces and Global Addresses," Internet-Draft, IETF Monami6 Working Group, November 2008.

[4] B. Radojevic and M. Zagar, "Analysis of issues with load balancing algorithms in hosted (cloud) environments," *Proceedings of the 34th International Convention MIPRO*, pp. 416–420, May 2011.

[5] L. Gharai, C. Perkins, and T. Lehman, "Packet reordering, high speed networks and transport protocol performance," *IEEE International Conference on Computer Communications and Networks (ICCCN)*, pp. 73–78, October 2004.

[6] D. Kaspar, K. Evensen, A. Hansen, P. Engelstad, P. Halvorsen, and C. Griwodz, "An analysis of the heterogeneity and IP packet reordering over multiple wireless networks," *IEEE Symposium on Computers and Communications (ISCC)*, pp. 637–642, July 2009.

[7] M. Przybylski, B. Belter, and A. Binczewski, "Shall we worry about packet reordering," *Computational Methods in Science and Technology*, vol. 11, no. 2, pp. 141–146, January 2005.

[8] S. Tinta, A. Mohr, and J. Wong, "Characterizing End-to-End Packet Reordering with UDP Traffic," *IEEE Symposium on Computers and Communications*, pp. 321–324, July 2009.

[9] S. Spirou, "Packet Reordering Effects on the Subjective Quality of Broadband Digital Television," *IEEE Tenth International Symposium on Consumer Electronics (ISCE)*, pp. 1–6, September 2006.

[10] Y. Ito, S. Tasaka, and Y. Fukuta, "Psychometric analysis of the effect of end-to-end delay on user-level QoS in live audio-video transmission," *IEEE International Conference on Communications*, vol. 4, pp. 2214–2220, June 2004.

[11] C.-C. Lin, H.-H. Chin, and D.-J. Deng, "Dynamic Multiservice Load Balancing in Cloud-Based Multimedia System," *IEEE Systems Journal*, vol. 8, no. 1, pp. 225–234, March 2014.

[12] G. Detal, C. Paasch, S. Van Der Linden, P. MéRindol, G. Avoine, and O. Bonaventure, "Revisiting Flow-based Load Balancing: Stateless Path Selection in Data Center Networks," *Computer Networks*, vol. 57, no. 5, pp. 1204–1216, April 2013.

[13] S. Prabhavat, H. Nishiyama, N. Ansari, and N. Kato, "On Load Distribution over Multipath Networks," *IEEE Communications Surveys Tutorials*, vol. 14, no. 3, pp. 662–680, March 2012.

[14] S. Kandula, D. Katabi, S. Sinha, and A. Berger, "Dynamic Load Balancing Without Packet Reordering," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 2, pp. 51–62, March 2007.

[15] J.-O. Kim, P. Davis, T. Ueda, and S. Obana, "Splitting downlink multimedia traffic over WiMAX and WiFi heterogeneous links based on airtime-balance," *Wireless Communications and Mobile Computing*, vol. 12, no. 7, pp. 598–614, July 2010.

[16] F. Zhong, C. K. Yeo, and B. S. Lee, "Adaptive load balancing algorithm for multiple homing mobile nodes," *Elsevier Journal of Network and Computer Applications*, vol. 35, no. 1, pp. 316 – 327, January 2012.

[17] L. Shi, B. Liu, C. Sun, Z. Yin, L. Bhuyan, and H. Chao, "Load-Balancing Multipath Switching System with Flow Slice," *IEEE Transactions on Computers*, vol. 61, no. 3, pp. 350–365, March 2012.

[18] O. Delgado and F. Labeau, "Uplink Load Balancing over Multipath Heterogeneous Wireless Networks," *IEEE 81st Vehicular Technology Conference (VTC Spring)*, pp. 1–6, May 2015.

[19] S. Prabhavat, H. Nishiyama, N. Ansari, and N. Kato, "Effective Delay-Controlled Load Distribution over Multipath Networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 22, no. 10, pp. 1730–1741, October 2011.

[20] M. Li, H. Nishiyama, N. Kato, K. Mizutani, O. Akashi, and A. Takahara, "On the fast-convergence of delay-based load balancing over multipaths for dynamic traffic environments," *Wireless Communications Signal Processing*, pp. 1–6, October 2013.

[21] K. Hashiguchi, Y. Kon, M. Hasegawa, K. Ishizu, and H. Harada, "Traffic allocation control using support vector machine in heterogeneous wireless link aggregation," *IEEE Consumer Communications and Networking Conference*, pp. 653–657, January 2011.

[22] J. Wu, J. Yang, Y. Shang, B. Cheng, and J. Chen, "SPMLD: Sub-packet based multipath load distribution for real-time multimedia traffic," *Journal of Communications and Networks*, vol. 16, no. 5, pp. 548–558, October 2014.

[23] *e-Handbook of Statistical Methods: Single Exponential Smoothing*. NIST/SEMATECH at the National Institute of Standards and Technology, April 2012.

[24] M. Caramia and P. Dell'Olmo, *Multi-objective Management in Freight Logistics Increasing Capacity, Service Level and Safety with Optimization Algorithms*. Springer, 2008.

[25] L. N. Vicente and P. H. Calamai, "Bilevel and multilevel programming: A bibliography review," *Journal of Global Optimization*, vol. 5, no. 3, pp. 291–306, February 1994.

[26] A. A. Pessoa, M. Poss, and M. C. Roboredo, "Solving bilevel combinatorial optimization as bilinear min-max optimization via a branch-and-cut algorithm," *Annals of XLV Brazilian Symposium of Operations Research*, vol. 2, pp. 2497–2508, September 2013.

[27] B. Colson, P. Marcotte, and G. Savard, "An overview of bilevel optimization," *Annals of Operations Research*, vol. 153, no. 1, pp. 235–256, April 2007.

[28] J. Kleinberg and E. Tardos, *Algorithm Design*. Pearson Education, 2006.

[29] G. Attiya and Y. Hamam, "Two phase algorithm for load balancing in heterogeneous distributed systems," *IEEE 12th Euromicro Conference on Parallel, Distributed and Network-Based Processing*, pp. 434–439, February 2004.

[30] S. Tomar, "Converting video formats with ffmpeg," *Linux journal*, vol. 2006, no. 146, p. 10, June 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1134782.1134792

[31] W. Zhang and J. He, "Modeling End-to-End Delay Using Pareto Distribution," *Second International Conference on Internet Monitoring and Protection (ICIMP)*, pp. 21–21, July 2007.

[32] L. Lam, K. Su, C. Chan, and X. Liu, "Modeling of round trip time over the internet," *Asian Control Conference*, pp. 292–297, August 2009.

[33] A. Jayasumana, N. Piratla, T. Banka, A. Bare, and R. Whitner, "Improved Packet Reordering Metrics," Internet Requests for Comments, RFC 5236, June 2008.

[34] B. Ye, A. Jayasumana, and N. Piratla, "On Monitoring of End-to-End Packet Reordering over the Internet," *International Conference on Networking and Services (ICNS)*, pp. 3–3, July 2006.

[35] J. Klaue, B. Rathke, and A. Wolisz, "EvalVid-A framework for video transmission and quality evaluation," *In Proceedings of 13th International Conference on Modelling Techniques and Tools for Computer Performance Evaluation*, pp. 255–272, September 2003.

Oscar Delgado received the Ph.D. degree in electrical engineering from McGill University, Montreal, QC, Canada, in 2016, and the M.A.Sc. degree from Concordia University, Montreal, QC, Canada, in 2010. His industrial experience include a network engineering position in the R&D department with Telefonica, Lima, Peru. His current research interests are in applications of 5G wireless mobile communication technologies, including software defined networks, network virtualization and green wireless systems. His focus is on the analysis and design of video traffic management techniques, resource allocation strategies, and energy efficiency algorithms.

Fabrice Labeau is an Associate Professor with the Electrical and Computer Engineering Department, McGill University, where he holds the NSERC/Hydro-Québec Industrial Research Chair in Interactive Information Infrastructure for the Power Grid. From 1996 to 2000, he was with the Communications and Remote Sensing Laboratory, Université Catholique de Louvain (UCL), Belgium, where he received his Ph.D. degree. His research interests are in applications of signal processing to healthcare, power grids, communications and signal compression. He has authored more than 175 refereed papers in refereed journals and conference proceedings in these areas.

He is the associate director for operations of SYTACom, an interuniversity research center grouping 50 professors and 500 researchers from 10 universities in the province of Quebec, Canada. He has held several administrative and management positions at McGill University, including associate department Chair, Associate Dean, Interim Chair and Acting Dean. He is currently Associate Dean for Faculty Affairs in McGill's Faculty of Engineering. He is junior past president of the IEEE Vehicular Technology Society, President-Elect of the IEEE Sensors Council, and the past chair of the Montreal IEEE Section. He received in 2015 the McGill University Equity and Community Building Award (team category), as well as the 2008 and 2016 Outstanding Service Award from the IEEE Vehicular Technology Society.