A Load Balancing Solution for Improving Video Quality in Loaded Wireless Network Conditions

Adriana Hava, *Member, IEEE,* Yacine Ghamri-Doudane, *Member, IEEE,* John Murphy, *Senior Member, IEEE,* and Gabriel-Miro Muntean, *Senior Member, IEEE*

***Abstract*—There is an increasing user demand for high-quality content-rich multimedia services. Despite their advantages, current wireless networks in general and wireless mesh networks in particular have limitations in terms of Quality of Service (QoS) provisioning, especially when dealing with increased amounts of time sensitive traffic such as video.**

This paper presents ViLBaS, a load balancing-based mechanism which enhances delivery performance of video services over multi-hop wireless mesh networks and improves user Quality of Experience (QoE) levels. ViLBaS involves performance monitoring at the level of the nodes and load balancing by off-loading traffic from loaded nodes to less loaded neighbouring nodes.

Simulation-based results show how the proposed ViLBaS improves video delivery performance in terms of both QoS and QoE metrics (delay, throughput, packet loss and Peak Signal-to-Noise-Ratio (PSNR)). The comparison is performed against three other traditional approaches in different network topologies, for diverse video flow distributions, and different sizes for the video queue.

I. Introduction

Wireless networks are gaining popularity and represent an attractive solution for providing network connectivity, including for delivery of rich media time-sensitive traffic. Diverse networked clients can generate and consume various types of traffic, which impact differently on network performance, from lightweight web-browsing data traffic to heavy traffic, such as video.

Besides the many features that make wireless networks a very good option for real life deployments and support for high bitrate traffic distribution, they also face multiple challenges.

One of the main challenges for wireless networks is the continuous increase of traffic, mostly due to the increase in user demand for high-quality multimedia content. This is as many top market players such as Cisco predict that video traffic will continue to be the largest contributor to the overall data traffic and will account for more than 82% of the traffic in 2021 [1]. In this context, providing high QoS levels [2] and consequently high perceived quality to end-users [3] is essential for the success of wireless networks in general, wireless mesh networks in particular and especially their video [4] or rich media services [5].

A. Hava and J. Murphy are with Performance Engineering Laboratory, School of Computing, University College Dublin, Ireland (email: j.murphy@ucd.ie

Y. Ghamri-Doudane is with L3i, University of La Rochelle, France (email: yacine.ghamri@univ-lr.fr)

G.-M. Muntean is with Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Ireland (email: gabriel.muntean@dcu.ie).

The European Union funding for the NEWTON project under the Horizon 2020 Research and Innovation Programme with Grant Agreement no. 688503 is gratefully acknowledged (http://newtonproject.eu).

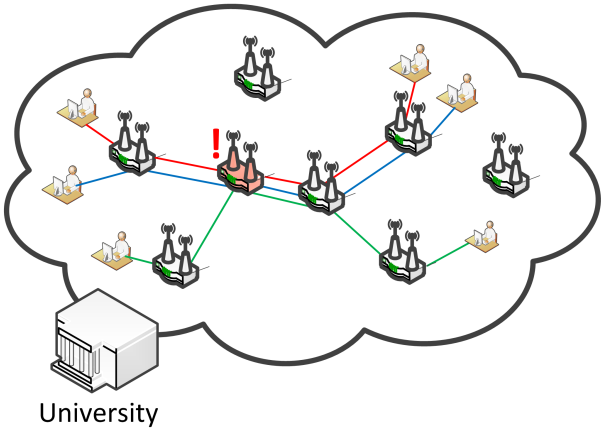

Fig. 1: Content distribution over a university WMN

Figure 1 illustrates a Wireless Mesh Network (WMN) deployed at the level of a university campus, which supports diverse content delivery, including rich media. The WMN consists of wireless interconnected routers (depicted in the figure as white boxes with two antennas), with limited or no mobility, arranged in a mesh topology. These mesh routers forward data traffic in a multi-hop fashion to and from mesh client devices [6]. These devices can be either stationary or mobile. A possible scenario involves many students accessing various video learning materials over the WMN, putting pressure on the network. An example of video traffic distribution between WMN clients is depicted in the figure. The coloured lines illustrate possible traffic patterns between mesh routers.

Most WMNs rely on Wi-Fi-based unlicensed spectrum, such as IEEE 802.11, and therefore they inherit many disadvantages of this technology, such as lack of support for QoS provisioning. To this contributes the fact that in a WMN the traffic is not evenly distributed, resulting in some nodes being heavily loaded (e.g. the mesh node highlighted with red in Figure 1), while others carry a small amount of traffic only, or no traffic at all (such as the wireless mesh node at the top of the figure).

In these conditions, besides increased delays, there is also an increase in packet loss, influencing data transmissions in general and severely affecting any video delivery quality

in particular. Indeed, a node forwarding a large number of packets on a loaded path will get easily congested, the packets will buffer in the transmission queue and the delays will continue to increase. Moreover, once the node's queue is full, packets are dropped negatively affecting the service quality level. For video traffic distribution, early detection of node congestion and performing traffic load-balancing is highly important [7], particularly across a mesh network.

This paper introduces ViLBaS, a **Vi**deo **L**oad **Ba**lancing **S**olution for WMNs. ViLBaS targets the video flows inside a mesh network, and improves the QoS levels for client video delivery services. ViLBaS runs on top of any routing protocol, which performs neighbour discovery and distribution of neighbour information. ViLBaS identifies loaded nodes, re-routes traffic around them and balances the video traffic.

Alternative load balancing solutions focus more on balancing the traffic load regardless of its type and do not consider QoS enhancement as core of the proposal. Unlike them, ViLBaS includes the following innovative aspects:

- Delay-sensitive traffic (e.g. video) is prioritized over data traffic.
- Congestion is detected by monitoring the queue occupancy levels at network interface cards.
- Video flows are re-routed individually, and this decision is triggered by the congested node.
- Load-balancing is performed such as to increase user QoS levels.

We compare the performance of the ViLBaS mechanism with that of existing state of the art routing algorithms, employing metrics such as hop-count and a metric proposed by De Couto [8], and a solution in which no load balancing is used. The results obtained from simulations show that ViLBaS outperforms the other routing solutions in terms of various QoS metrics (delay, throughput, packet loss) and estimated user perceived video quality.

The remainder of the paper is organized as follows: Section II presents an overview of relevant existing routing metrics and load balancing solutions proposed in the literature. Section III describes in details the proposed ViLBaS algorithm for load-balancing in a mesh network. Section IV presents the simulations carried to demonstrate the improvements brought by our mechanism, along with an analysis of the results. Section V discusses diverse general aspects and limitations of the proposed solution, and section VI includes the conclusions of our work and presents our future research plans.

II. Related Works

In wireless networks context, research has focused on proposing solutions to improve diverse performance-related parameters including QoS and energy consumption [9]. However related to load-balancing-aware routing solutions, this section summarizes diverse metrics used for routing in WMNs and discusses some load-balancing solutions.

Routing metrics are used by routing protocols in the process of route discovery and route decision in relaying packets. They are employed to help a source node identify the best route among multiple different paths to a destination.

The hop-count is the most widely used metric by routing protocols, including Optimized Link State Routing (OLSR) [10] in their quest to find the shortest route between source and destination nodes. However, as using hop-count does not consider specific characteristics of wireless mesh environments, including the fact that the links are error-prone [11], it provides poor QoS support to multimedia traffic [12].

A metric proposed by De Couto [8] attempted to consider link quality at MAC layer between nodes, but traffic load was not considered.

The *Estimated Transmission Time (ETT)* metric considers link quality in terms of link transmission rate and packet size. However, similar to De Couto's metric, ETT does not consider node load.

The *Weighted Cumulative ETT (WCETT)* metric [13] enhanced ETT by considering intra-flow interference in multi-radio WMNs. WCETT favours paths with low number of nodes that transmit on the same channel, thus reducing the intra-flow interference. However, WCETT still does not consider traffic load. Additionally WCETT solutions are affected by the non-isotonicity property which makes them unusable for proactive routing protocols [14].

Although the above discussed metrics work well for delivering data in WMNs, they are traffic-agnostic, and therefore they cannot guarantee high QoS levels for the multimedia content delivered. A more detailed discussion about routing metrics can be found in [15].

In addition to the above presented metrics, which aim at finding high-throughput paths, other research papers focused on enhancing them by adding link-load knowledge.

The *WCETT-LB* [16] enhances WCETT routing metric by introducing a load-balancing component at the mesh routers and supporting a global load-aware routing. The load-balancing component takes into consideration the congestion level, considering the average queue length, and the traffic concentration level at each node. The authors consider that if the average queue length at a node is higher than a threshold, then the path is heavily loaded. The presented solution is compared only against the hop-count metric and the authors omit to detail some aspects of their mechanism, such as the way the best available route is computed. Other missing pieces of information are threshold values used (congestion level threshold and threshold for path switching) to enable the mechanism, and how these values are determined. A drawback of the mechanism is the fact that the source nodes are the ones taking the decision to switch the paths for their traffic. Hence, is likely all the traffic from the congested node to be re-routed and leave the node traffic-free, and thus simply moving the congestion to other nodes. The results show how packet delivery ratio drops significantly (e.g. to 50% packet loss) and how loss increases with simulation time.

A load-balancing solution based on a neighbourhood load routing metric is presented in [17]. The authors aim

at bypassing the loaded node's neighbourhood instead of avoiding only the loaded node. Hence, the authors propose and use the Neighbourhood Load Routing (NLR) metric which computes the average load of the neighbourhood of a link. The NLR metric considers three aspects when selecting a path: network interface queue length, neighbourhood interference and neighbourhood bandwidth. The main draw-back of this mechanism is that it is very likely for a mechanism employing this metric to route the traffic on the edges of the mesh network and hence, move the congestion on the edge links.

Comprehensive studies about load-balancing routing protocols exist in the literature including in the area of mobile ad-hoc networks. In [18] the authors discuss various load metrics and categorise the routing protocols based on the load-balancing technique used. However, the authors analysis does not consider how these metrics behave in the presence of video traffic.

A routing metric based on the capacity estimation of the WMN nodes is proposed in [19]. The metric adapts based on the estimated load of the mesh nodes, thus optimizing the overall mesh network's load. However, this requires constant monitoring of the mesh nodes and identifying complete new routes through the periodic AODV discovery process, which can lead to switching video flows to a new path even when this is not necessary. On the other hand, our solution is triggered by a mesh node before the congestion occurs and the re-routing is done locally.

An important related work is our previous approach [15],[20] which focuses on improving QoS levels for video deliveries in WMNs by employing a distributed load-balancing solution. The solution is evaluated by deploying real video traffic over an emulated WMN test-bed and using objective video quality assessment tools. The work presented in this paper is a step forward and describes in details the methodology for selecting the queue occupancy threshold that triggers the load balancing mechanism, and presents a thorough solution assessment in various topology scenarios.

III. Video Load Balancing Solution for WMNs (ViLBaS)

A. Overview

Considering typical WMN random distribution of video traffic, a routing protocol, such as for instance OLSR, distributes the traffic flows on the shortest path between the source and destination. This may create highly congested nodes, which carry large numbers of video flows, affecting their quality. In this context ViLBaS is introduced on top of the routing protocol to balance the video flow distribution inside a mesh network, in order to reduce congestion on the major data delivery paths and eventually increase delivered video quality levels.

The routing protocol i.e. OLSR performs network discovery and fills the routing tables for all mesh nodes. It finds the shortest route between all source and destination pairs in the mesh network. For instance the hop-count field in the OLSR messages contains the number of hops the packet traveled within the network, as described in RFC 3626 [10] and this field is used by ViLBaS to identify the shortest distance between all nodes in the mesh network. Within the network topology, OLSR exchanges periodically information among nodes, and every node maintains info on the whole network topology. This behaviour can be used for detecting new nodes that joined the network, or node failures. Routing table calculation is performed by ViLBaS.

Acronym	Type	Priority	
AC_BK	Background Traffic	4	Lowest
AC_BE	Best-Effort Traffic	3	—
AC_VO	Voice Traffic	2	—
AC_VI	Video Traffic	1	Highest

TABLE I: IEEE 802.11e Classes of Traffic

ViLBaS considers different classes of traffic, making use of the QoS traffic class prioritisation mechanism introduced first by IEEE 802.11e [21]. This mechanism prioritizes time-sensitive traffic, such as voice or video, and gives lower priority to traffic from best effort or other data exchanging applications. This is achieved by associating priority queues to different traffic classes. Four such queues are defined in IEEE 802.11e, each having a different priority as presented in Table I.

Each arriving packet is inserted in the queue allocated to its type. So, the packets from AC_VO and AC_VI queues wait a shorter time before being transmitted compared to the packets stored in AC_BE and AC_BK queues.

ViLBaS focuses on the AC_VI queue and targets video flows inside wireless mesh networks. It identifies loaded nodes, re-routes traffic around them and balances the load at the same time.

B. Architecture Description

Figure 2 illustrates the ViLBaS node-level architecture comprising of components, which reside at the data-link layer for identifying node congestion and for flow selection, and at the network layer for congestion notification and new route selection.

ViLBaS is a distributed solution residing on each node participating in the mesh network, and is composed of the following major components:

① Node Activity Detector,
② Flow Selector,
③ Previous Node Identifier, and
④ New Route Selector.

Information about node congestion is gathered by the ① *NodeActivityDetector* component at the data-link layer from each video-queue interface. At this layer, knowledge about the flows running through the nodes is available. The selection of the flow which will be re-routed is also done at this level through the ② *Flow Selector* component. This information is passed to the *Routing Protocol* component located at the network layer. Having this information available, the ③ *Previous Node Identifier* component is able to determine the previous node on the

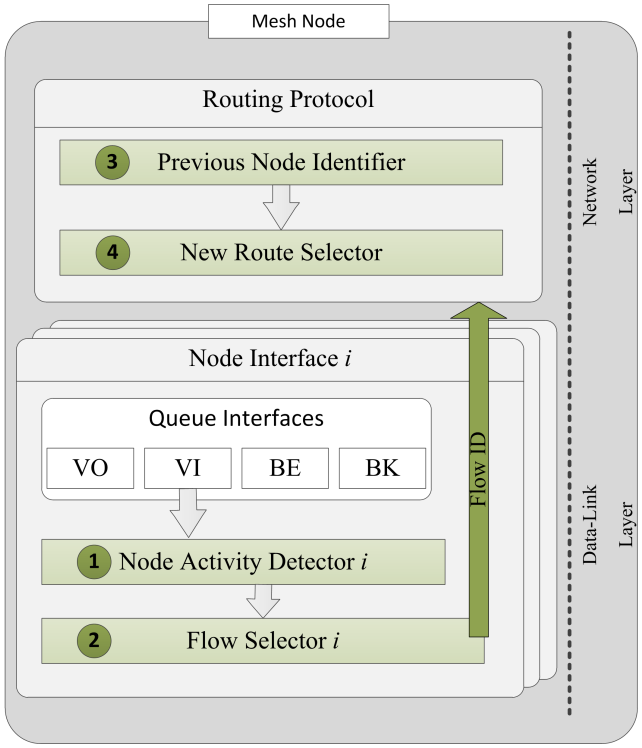

Fig. 2: ViLBaS Node-Level Architecture

Symbol	Definition
LN	Loaded Node
NN	Neighbour Node
PN	Previous Node
N_{MN}	List of neighbours for mesh node
F_{sel}	Selected Flow
L_F	List of flows running on a node
R	New Constructed Route
MN_c	Current Mesh Node
MN_d	Destination Mesh Node
$VIQO$	Video Queue Buffer Occupation

TABLE II: ViLBaS's Parameters

path of the *Flow ID*. The ④ *New Route Selector* component will thus identify the new route which bypasses the loaded node. These components are described in more detail in the following subsections. The parameters used to describe the algorithms associated with these components are summarised in Table II.

C. Node Early Congestion Detection

As mentioned before, the traffic in a WMN is more likely to be unevenly distributed across the network. In some situations, some nodes carry more traffic and serve more flows than other nodes which are less loaded. The role of the *Node Early Congestion Detection* component is to identify when a node becomes congested. The node is considered congested when the queue storing the video packets (AC_VI) reaches a certain threshold.

The mechanism that triggers the execution of ViLBaS is presented in Algorithm 1. The algorithm runs on each node

Algorithm 1: Node Early Congestion Detection Algorithm

Data: AC_VI_i
Result: LN
foreach AC_VI *at* MN_c **do**
  Compute $VIQO_{AC\_VI}$ at MN_c;
  if *((*$VIQO_{MN_c}$ *) and (T elapsed))* **then**
    == MN_c's congestion level exceeded;
    MN_c becomes LN;
    return LN;
  else
    == MN_c is below threshold;
    return *null*

Algorithm 2: Flow Selector Algorithm

Data: $L_F = < F_1; F_2; :::; F_i; :::; F_n >$
Result: F_i
foreach F_i 2 L_F **do**
  $n_{F_i} = F_i$'s share of the video queue;
return F_{sel} having the largest n_{F_i};

and it is executed for each incoming packet at every video queue of every interface (AC_VI_i).

Incoming video packets are stored in the forwarding video queue of each node on the path of the video flow. When the number of packets that exit the queue is smaller than the number of packets entering in the queue, the queue will get filled. In order to detect this situation that eventually leads to dropped packets, each mesh node monitors the amount of packets enqueued in the AC_VI queue. $VIQO_{MN}$ represents the number of packets stored in the video queue of the mesh node MN. Once a certain threshold value is reached, the mesh node, MN, is considered at risk to become a loaded node (LN). This means that the AC_VI queue is likely to fill soon, leading any incoming packets to be dropped. Once, MN becomes a LN, then it selects a flow, F_i, that must be re-routed by the previous node around the loaded one. This is performed by the *Flow Selector* component described in the next subsection.

The *Node Early Congestion Detection* component automatically triggers ViLBaS re-routing process once the threshold is reached. However, in order to avoid excessive re-routing a back-off period of T is considered.

D. Flow Selector

In this paper, a video traffic flow is defined as a sequence of packets to be enqueued in the AC_VI queue with the same IP source and destination pair.

The *Flow Selector* component selects a flow to be removed and re-routed through other nodes. Therefore, the new route for the selected flow must avoid the respective node on its new path. In this selection process, a suboptimal option is to let the node to select a random flow. Unlike

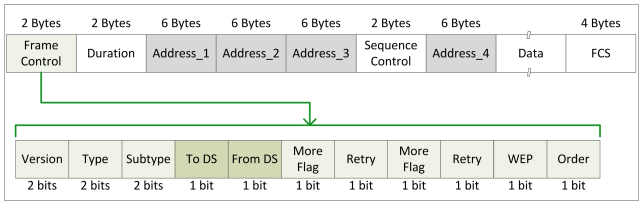

Fig. 3: IEEE 802.11 MAC Header

To DS	From DS	Address_1	Address_2
0	0	Destination	Source

TABLE III: Address Fields in IEEE 802.11 MAC Header

previous works carried in the area of traffic load-balancing, we propose a flow selection mechanism (Algorithm 2), which performs as follows.

Each mesh node keeps locally a list of the flows that pass through it, named L_F. When the node's threshold, , is exceeded, a flow from the list will be selected for re-routing. In this case, the flow which occupies the largest share of the video queue in the loaded node is selected for re-routing. The packets belonging to the selected flow, F_{sel}, already enqueued in LN's AC_VI queue will still be transmitted to their destination.

L_F is updated every time a new packets is received in the AC_VI queue. However, if no packets belonging to a certain flow arrive for a period P of time, the flow is considered inactive on that node and is removed from L_F.

E. Previous Node Identifier

The selected flow re-routing mechanism, which is detailed in the next section, III-F, starts at the previous node. The previous node is identified based on the information stored in the IEEE 802.11 MAC header of the packets received and on the ARP table of the node.

The IEEE 802.11 MAC header (depicted in Figure 3) may contain up to four address fields, opposed to the IP header which contains only two address fields. The fields of interest of the MAC header are Address_1, Address_2, coloured in grey in Figure 3. These addresses are in the 48-bit format and carry different meaning based on the **To DS** and **From DS** bits (coloured in dark green) in the Frame Control field. In a mesh network scenario the packets are exchanged between stations, thus the **To DS** and **From DS** fields are set to 0 and the four addresses of the MAC header are filled according to Table III. The Destination represents the MAC address of the node who just received the packet, the Source represents the MAC address of the node who sent the packet.

The ARP table of the node receiving the packets matches the source MAC address to the IP of the node sending the data. In this way, a node always knows the IP of the previous node who sent the packet to it. Thus, for any video flow a node can identify the IP of the previous node sending a packet belonging to a flow.

Algorithm 3: New Route Selector Algorithm

```
1  FormRoute(R, MN_c, MN_d);
2  U_max = MaxValue;
3  if (MN_c == MN_d) then
4  |  return R
5  else
6  |  foreach NN 2 N_MN_c do
7  |  |  Calculate U_MN_c;NN (Eq. 1);
8  |  |  if (NN == MN_d) then
9  |  |  |  R   NN;
10 |  |  |  return FormRoute(R, NN, MN_d);
11 |  |  if (U_MN;NN   U_max ) then
12 |  |  |  U_max = U_MN;NN;
13 |  |  |  NN_max = NN;
14 |  R = R   NN_max;
15 |  return FormRoute(R, NN, MN_d);
```

F. New Route Selector

The role of the *New Route Selector* component is to identify a new route, R, for the flow. Once a flow is selected by the *Flow Selector* component to be re-routed from the LN, the next step is to identify a new route for it. Next, we present a route selection algorithm for the selected flow for re-routing. The route calculation starts at the previous node, PN, and it tries to find the shortest route in terms of queue occupancy and hop-count to destination. The identification of the PN is done by the *Previous Node Identifier* component. The newly proposed recursive route finding algorithm is called `FormRoute` and is presented in Algorithm 3.

Each node will look at its neighbour nodes and will select the next hop neighbour based on a utility function $U_{MN;NN}$. The utility function is computed based on a summative weighted method, which has two terms: the traffic load of the neighbour node and the distance to the destination of the flow. The overall utility function is defined in Eq. (1):

$$U_{MN;NN} = \quad U_{VIQO_{NN}} + (1 \quad) \; U_{DNN;DN} \qquad (1)$$

The term $U_{VIQO_{NN}}$ is a utility function defined for node occupancy and $U_{DNN;DN}$ is a utility function defined for the distance to destination. is the weight giving more importance to one or the other element of the sum. Each utility term is described below in detail. $U_{VIQO_{NN}}$, named *Queue Occupancy Utility Function*, represents the amount of packets stored in the queue and is intended to be as small as possible. In order to calculate the queue occupancy utility function, the value representing the current number of packets stored in the video queue is divided by the maximum capacity of the queue as shown in Eq. (2).

$$U_{VIQO_{NN}} = \frac{VIQO_{NN}}{Max_{VIQO}} \qquad (2)$$

The element $U_{DCN;NN}$, named *Destination Distance Utility Function*, maps the distance from the neighbour node to the destination point as in Eq. (3). $MaxDist$ is the maximum acceptable number of hops for which the video quality does not degrade (i.e. 10 hops). Hence, to obtain $U_{DCN;NN}$ we divide the number of hops between the neighbour node and destination to $MaxDist$.

$$U_{DNN;DN} = \frac{D_{NN;DN}}{MaxDist} \tag{3}$$

The term balances the effect of the two terms within the weighted utility function employed for re-routing. Higher value of results in higher importance be given to the neighbour node traffic, whereas a lower value for puts emphasis on finding a re-routing solution with shorter distance to the flow destination. The former approach may result in re-routing over less loaded, but longer paths, whereas the latter in shorter re-routing paths, but which are still affected by traffic load.

As for both, $U_{MN;NN}$ and $U_{DCN;NN}$, the denominator is smaller than the numerator, the two utility functions are ranging between 0 and 1 and have no measurement unit. Hence, they can be used in Eq. (1).

The utility function in Eq. (1) is computed for each neighbour node in turn. If a mesh node (MN_c) has a one-hop link with the destination node (MN_d) then the algorithm stops and adds the destination node to the route R, otherwise, the computed utility function is compared against the best value discovered among all neighbours. If the newly calculated value is better than U_{max}, the coresponding node (NN_{max}) is the best neighbour and it is selected. These steps are repeated for each node until the destination node is reached.

The *New Route Selector* algorithm relies on a series of light-weight messages exchanged between the nodes. The first message is exchanged between the loaded node and the previous node on the selected path. The loaded node informs the previous node that it is selected. The previous node and all the nodes on the new route exchange messages only with their one-hop neighbour, making the message exchange localised.

This process is illustrated in the example given in Figure 4 through circled numbers. A video flow, represented by a red line, is running between node 0 and node 15 in a sixteen-node grid deployment.

① A Loaded Node (LN) (i.e. node 6) informs the Previous Node (PN) (i.e. node 5) on the path of the red flow that it must re-route this flow.

② PN identifies one hop neighbours from its routing table.

③ PN requests information (video queue occupancy and distance to destination) from its neighbours.

④ The requested information is replied.

⑤ PN computes $U_{MN;NN}$ for each neighbour and selects the one with the highest $U_{MN;NN}$ (i.e. node 9).

⑥ PN updates its routing table to point to the selected neighbour instead of LN.

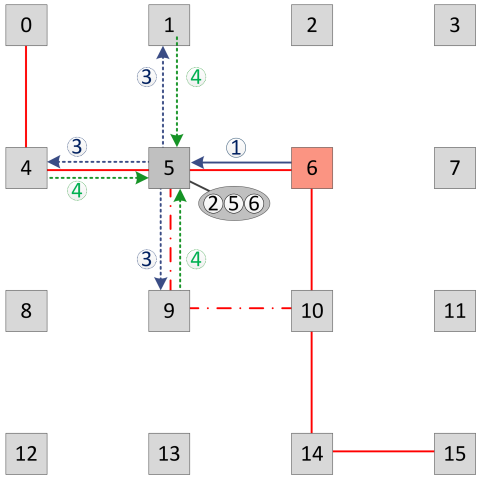

Fig. 4: Message Exchange between Mesh Nodes

Algorithm 4: ViLBaS Mechanism

Data: Mesh Nodes
Result: Load-Balanced Video Traffic
1 LN = Node Early Congestion Detection;
2 F_{sel} = FlowSelectionOnLoadedNode(L_F);
3 Identify previous node PN for F_{sel};
4 R = RouteSelector;
5 Update Routing Table on each MN 2 R;

These steps are repeated until the destination node is reached. Finally, the new route (dashed line) is bypassing the loaded node.

G. ViLBaS Solution

ViLBaS re-routes flows from highly congested nodes to less congested paths and as result, it improves QoS levels.

The proposed load balancing mechanism for improved video quality is depicted in Algorithm 4 as a pseudo-code which puts together the previous described algorithms. Each mesh node monitors continuously its own video queue occupancy. Once a specific threshold, , per node is exceeded, the respective node signals this event. The next step selects the flow to be removed from the loaded node and re-routed through a different path which avoids the congested node. Hence, the previous node for that flow is identified and the new path discovery is initiated on that node. The new route avoids the loaded node and selects nodes which have a low queue occupancy so it does not disrupt the traffic. When a route that matches this criteria is discovered, the routing table on each node on the path is updated accordingly.

IV. Performance Analysis

This section describes the simulation environment used and the performance evaluation carried to assess ViLBaS in comparison with alternative approaches.

TABLE IV: Video Statistics

Parameter	Value
Mean Frame Size	770 Bytes
Max Frame Size	6735 Bytes
Mean Bit Rate	150 kbps
Peak Bit Rate	800kbps

TABLE V: ViLBaS Simulation Parameters

Parameter	Value
α	0.5
T = Backoff Period	2 seconds
P = $\mathcal{L}_{\mathcal{F}}$ Refresh	1 second
τ = Queue Threshold	60%

A. Simulation Setup

ViLBaS has been deployed and assessed using the NS-3 network simulator, version 3.10 [22]. ViLBaS is built on top of OLSR, which performs network discovery and fills the routing tables for all mesh nodes.

ViLBaS is compared against OLSR when using the *hop-count* metric, *De Couto* metric, and, additionally a *static* routing table solution is also considered. For the *static routing* case, OLSR is used for initial route discovery and once the routes are discovered, OLSR is prevented from making route updates. *De Couto's metric* takes into consideration the link state between two nodes when finding the route to a destination. As the standard NS-3 version does not support this metric, the Luis da Costa Cordeiro NS-2 patch [23] was ported into NS-3.

Our simulation setup considers two topologies: a sixteen-node grid topology and a twenty-five-node grid topology, with nodes placed at 125 meters apart, as presented in Figure 5. These topology sizes and similar number of mesh nodes are used by many other researchers in their experiments [24, 25, 19]. The decision to use grid topologies is supported by a study on WMN performance. The study [26] has shown the benefits of grid topologies over random topologies in terms of coverage, connectivity and network throughput. However this does not limit the potential deployment of ViLBaS to other types of topologies. The communication between mesh nodes is using the IEEE 802.11a standard, while the communication between the clients and the mesh nodes is realized using the IEEE 802.11g standard. Setting the communication links in this way allows to avoid interferences. Because the mesh nodes are set 125 meters apart, we set the maximum channel capacity of a link inside the mesh network at 6 Mb/s.

The maximum size of the video queue is set to 50 packets for one set of simulations and 100 packets for the other set. This option is based on the legacy open source MadWifi drivers for Atheros chipsets (present on some wireless network interface cards) which use a driver ring buffer of 200 packets. The ath5k drivers for the same chipset divide these 200 packets equally among the four queues (VI, VO, BE and BK queues) [27]. Thus, we can assume the video queue can store 50 packets if the 200 packets are equally distributed. As well, this distribution can be uneven, by defining a larger video queue, e.g. 100 packets, and smaller queue sizes for the other queues. However, setting a video queue bigger than 100 packets leads to unusually large packet delays, so we decide to avoid such settings. Similarly, a video queue smaller than 50 packets seems to us unrealistic as it may lead to important packet loss rates.

TABLE VI: Simulation Setup

Parameter	Value
Simulator	NS-3.10 [22]
Topology	Grid 4x4 and Grid 5x5
Distance between nodes	125 m
WiFi Mesh Mode	IEEE 802.11a
Wifi Client Mode	IEEE 802.11g
WiFi Data Rate	6 Mbps
Network Access Method	CSMA-CA
Propagation Model	LogDistancePropagation LossModel
Error Rate Model	YansErrorRateModel
Remote Station Manager	ConstantRateWifiManager
Video Queue Size	50 and 100 packets
Traffic Type	MPEG4 Video Trace Files
Video Type	Medium Quality
Mean Bit Rate	150 kbps
Routing Algorithm	OLSR
Number of simulation epochs	10

In order to simulate video traffic as real as possible, medium quality video traces of MPEG4 streams [28] are used. A video trace file includes the frame number, frame type and frame size and describes real video traffic. Five medium quality video flows are considered and randomly distributed between mesh nodes. A higher number of video flows would overload the network and lead to high packet loss. The simulations performed show an acceptable packet loss for five medium quality video flows. The characteristics of the chosen video are presented in Table IV. These values are typical for wireless video content deliveries in the scenario presented in the *Introduction*. An example of video flows distribution is depicted in Figure 5.

To ensure the accuracy of the results obtained, ten distinct simulation runs are performed using different seeds. All results in the paper were obtained by averaging over these ten runs. Ten seeds are chosen as this value is commonly used in other works involving multimedia traffic, such as [29], [30]. The values used for ViLBaS's parameters are presented in Table V. The term is the weight giving more importance to one or the other component of the utility function from Eq. (1). T is defined as a back-off period to avoid excessive re-routing. P is the period after which a flow is considered inactive on a node. is the queue occupancy threshold value which triggers ViLBaS.

Table VI summarises the network parameters used in our simulation environment. The inter-node distance is set to 125 meters and, the maximum data rate transmission of a link inside the mesh network is set to 6 Mbps.

B. Performance Metrics

For each simulation performed, four performance metrics are evaluated:

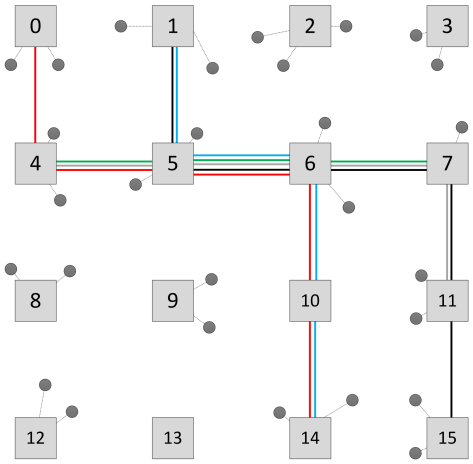

Fig. 5: Sixteen-Node Grid Simulation Scenario

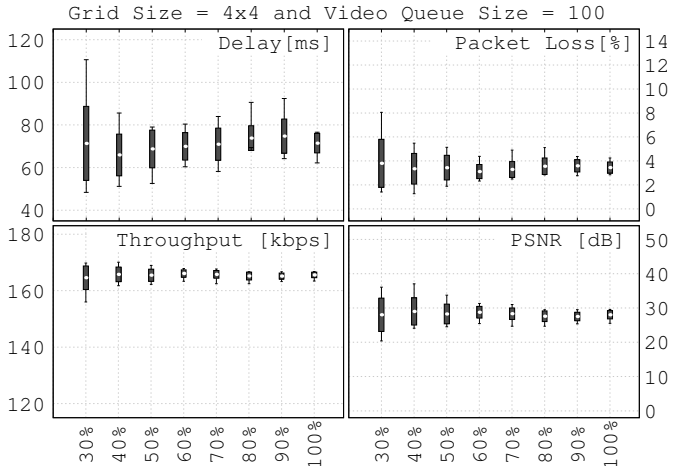

Fig. 6: Queue Occupancy Threshold Variation for a queue of 50 packets

Delay [ms] - The time needed for the packets to reach their destination;
Packet Loss [%] - The percentage of packets lost in relation to packets sent
Throughput [kbps] - The average network throughput;
PSNR [dB] - Widespread metric for measuring video quality. The PSNR value is calculated based on the loss and throughput rates using Eq. (4) from [31], where $MAX\_Bitrate$, EXP_Thr and CRT_Thr are maximum bitrate, expected and current throughput of the video stream, respectively.

$$PSNR = 20 \quad log_{10} \frac{MAX\_Bitrate}{\sqrt{(EXP\_Thr \quad CRT\_Thr)^2}} \tag{4}$$

C. Setting Queue Occupancy Threshold Value

As already presented in Section III, ViLBaS is triggered by the loaded node when the number of packets enqueued at the interface has reached a certain limit.

A set of simulations were setup to calculate the best value for the triggering threshold of the loaded node. The network is loaded with five video flows. Five was chosen as this number of flows loads the network and creates a high amount of packet loss. Then, for each set of simulations, the triggering threshold is varied for the queue occupancy from 30% up to 100%. Figure 6 and Figure 7 depict the evolution of the four metrics considered when we vary the mechanism's triggering-threshold along with their standard deviations, for a random distribution of video flows as presented in Figure 5 .

For each of the four metrics considered (delay, packet loss, throughput and PSNR) and for each triggering-threshold we plot a bar. The upper margin and lower margin of the black box represent the standard deviation around the average value. The average value is represented through a white dot, which is the middle of the box. The lower and upper whisker represent the minimum and the maximum

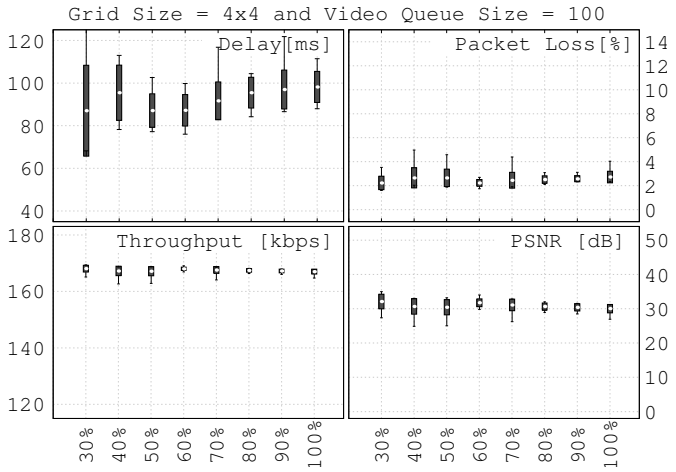

Fig. 7: Queue Occupancy Threshold Variation for a queue of 100 packets

value obtained, respectively. Each of the before mentioned values is computed as an average across the ten simulation and across all active video flows. In this way we show the performance of the overall mesh network.

Figure 6 and Figure 7 depict the results obtained for all four performance metrics (delay, packet loss, throughput and PSNR), when considering a queue length of 50 packets and 100 packets, respectively. Table VII shows the numeric results for the packet loss percentage for both queue lengths considered above. We justify that choosing a value of 60% for the queue occupancy is reasonable considering that it gives the lowest packet loss values (shown in the black cells of Table VII), which result in the highest PSNR values.

It can be observed, for both cases (when the queue can store 50 packets and when it can store 100 packets), that the 60% queue occupancy treshold obtains the lowest packet loss and the smallest standard deviation. This translates into a similar packet loss percentage across all deployed applications. For a threshold set to 30% queue occupancy, the variation of packet loss is significant for the case

TABLE VII: Packet Loss: Queue Size vs. Queue Occupancy Threshold

Queue Size	Queue Occupancy Treshold							
	30%	40%	50%	60%	70%	80%	90%	100%
50	3.80	3.35	3.44	3.11	3.29	3.56	3.59	3.45
100	2.23	2.65	2.65	2.23	2.45	2.51	2.57	2.72

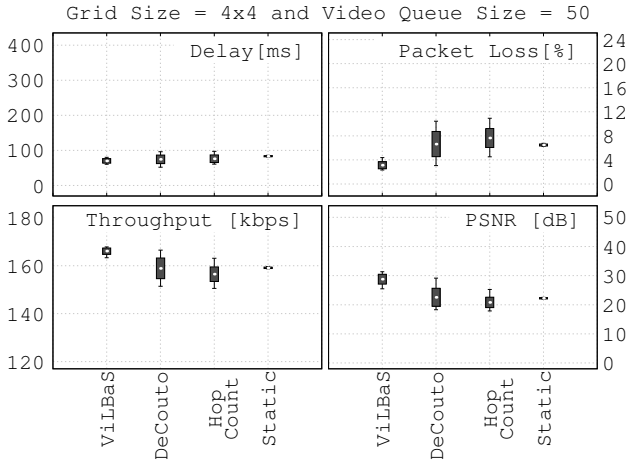

Fig. 8: ViLBaS performance for a queue size of 50 packets in a 4x4 grid topology

depicted in Figure 6 The trend observed for the packet loss is mirrored by the PSNR values obtained. If the mechanism is triggered at 60% queue occupancy, we obtain the highest PSNR value. This value decreases if the mechanism is applied when having a higher queue occupancy percentage (i.e. using a threshold value higher than 60%).

Regarding the delay measurements, it can be also observed that if ViLBaS is triggered at a higher queue occupancy threshold, the delay increases. This is expected to happen since packets are waiting in the queue for a longer time.

D. ViLBaS Performance Assessment

This section compares ViLBaS with the other three mechanisms mentioned earlier. We compare ViLBaS against:

Hop-count metric employed with OLSR - involves finding the shortest paths to destination,

De Couto metric employed with OLSR - involves finding high-throughput paths in multi-hop wireless networks,

Static routing - involves routes discovered initially which are kept static throughout the whole simulation, after the warm-up period.

Four scenarios are considered for evaluating the performance of ViLBaS. The scenarios considered analyze how

TABLE VIII: 4x4 Grid topology, Queue size = 50 packets

	Delay [ms]	Packet Loss [%]	Throughput [kbps]	PSNR [dB]	Decrease [%]
ViLBaS	69.96	3.11	166.08	28.77	—
De Couto	74.62	6.63	158.88	22.56	21
Hop Count	76.26	7.64	156.46	20.81	27
Static	83.36	6.48	159.11	22.20	22

TABLE IX: 4x4 Grid topology, Queue size = 100 packets

	Delay [ms]	Packet Loss [%]	Throughput [kbps]	PSNR [dB]	Decrease [%]
ViLBaS	87.22	2.23	167.99	31.74	—
De Couto	129.28	7.95	156.57	21.20	33
Hop Count	110.54	6.35	158.68	22.32	29
Static	142.06	5.82	160.44	23.12	27

ViLBaS performes for different network topologies and different buffering levels. These scenarios are:

16-node grid size & 50 packets queue size
16-node grid size & 100 packets queue size
25-node grid size & 50 packets queue size
25-node grid size & 100 packets queue size

For each scenario, ViLBaS is assessed with the other three solutions in terms of delay, packet loss, throughput and PSNR. The values obtained are presented in the following tables and figures. The last column of each table presents the improvement in terms of PSNR ViLBaS obtains when compared with the corresponding mechanism.

1) Sixteen-node Grid Topology: We first consider the case of a network with sixteen nodes arranged in grid topology. Figure 8 and Table VIII depict the results of the comparison when we use an interface queue size of 50 packets. As it can be observed, ViLBaS outperforms the other three metrics in terms of delay, packet loss, throughput and PSNR.

If we consider the delay performance metric, ViLBaS obtains the lowest overall delay and as well the standard deviation is the smallest when compared to the other mechanisms. This means that all the video flows running in the network are having low variations of delay between themselves. Compared to De Couto metric, ViLBaS improves the overall delay with 6%.

An important performance metric which has a big impact on the delivered video quality is packet loss. Loss values are presented in the top-right subgraph of Figure 8 and the second column in Table VIII. Note how ViLBaS gives the best results in terms of packet loss. For instance, compared to the OLSR with De Couto metric, ViLBaS decreases the packet loss with 53%.

When an interface waiting queue of 100 packets is considered, the results presented in Figure 9 and Table IX

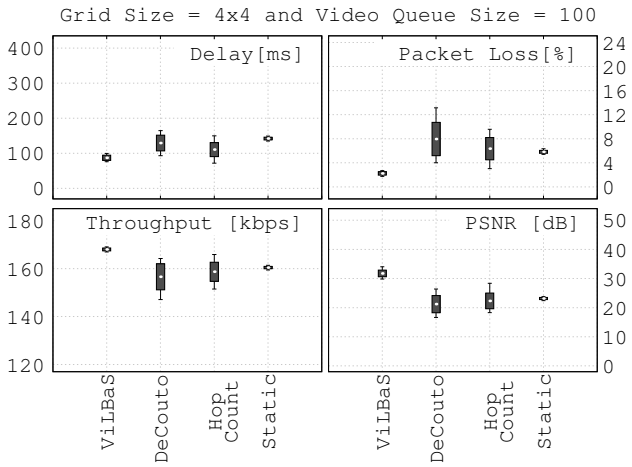

Fig. 9: ViLBaS performance for a queue size of 100 packets in a 5x5 grid topology

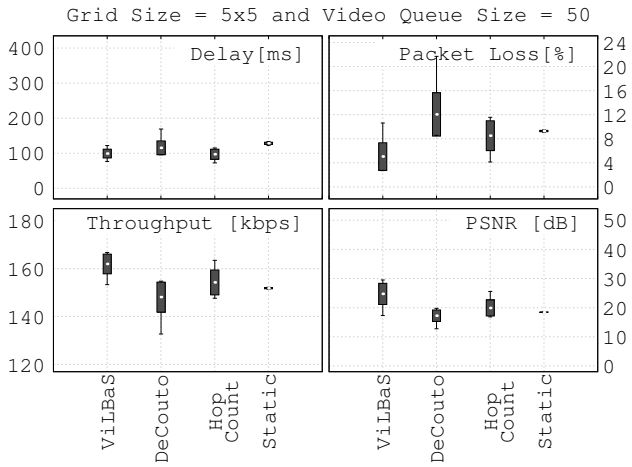

Fig. 11: ViLBaS performance for a queue size of 50 packets in a 5x5 grid topology

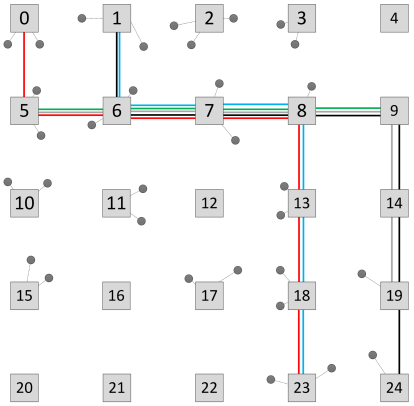

Fig. 10: Twenty-five-Node Grid Simulation Scenario

are obtained. As the interface's waiting queue can store a double number of packets compared to the previous case, the queueing delay increases accordingly. Hence, compared to the previous case, ViLBaS obtains an overall delay of 87.22 ms compared to 69.96 ms.

However, for a waiting queue of 100 packets (Table IX), the delay obtained by ViLBaS is 21% lower compared to the case when the hop-count metric was used and 32% lower compared with the case when De Couto metric was employed. Similarly, when comparing ViLBaS with the hop-count and De Couto metric cases, the standard deviations are much lower.

Analysing the packet loss, ViLBaS's loss is 2.23% only. This value is 72% smaller than the overall packet loss obtained when using OLSR with De Couto metric and 64% lower than using OLSR with the hop-count metric. This low packet loss is reflected into the values obtained for the PSNR as well. Compared to the other routing metrics, ViLBaS obtains the highest values for this metric: 31.74 dB, which is almost 33% higher then the corresponding value in the case of OLSR with De Couto metric.

2) Twenty five-node Grid Topology: A network with twenty-five nodes arranged in a grid topology is considered.

TABLE X: 5x5 Grid topology, Queue size = 50 packets

	Delay [ms]	Packet Loss [%]	Throughput [kbps]	PSNR [dB]	Decrease [%]
ViLBaS	99.24	5.04	161.94	24.72	—
De Couto	115.62	12.04	148.13	17.24	30
Hop Count	97.20	8.51	154.23	19.90	20
Static	128.58	9.27	151.83	18.44	25

An illustrative example of the flow distribution is presented in Figure 10. The source and destination for each flow are picked such as to resemble the sixteen-node grid topology, but scaled to a larger network.

Figure 11 and Table X depict the results of the comparison when we use an interface queue size of 50 packets. Again, ViLBaS outperforms all other solutions considered. In terms of delay, ViLBaS obtains an average delay of 99.24 ms, which is close to the one obtained using OLSR with hop-count: 97.20 ms. In terms of packet loss, ViLBaS's improvement is significant. The average packet loss is 5.04%, which is 40% lower than the value obtained by OLSR with hop-count. This low packet loss is reflected into the PSNR value, where ViLBaS gives an average PSNR value of 24.72 dB, which is the best value obtained across all metrics. It is almost 20% higher than the PSNR value obtained by OLSR employing the hop-count metric. The ViLBaS PSNR improvement during video delivery in a 5x5 grid topology is shown in the bottom-right subgraph of Figure 11 and PSNR column of Table X.

Figure 12 and Table XI compares the considered alternative solutions when we use an interface queue size of 100 packets. ViLBaS outperforms the three other schemes in this case as well in terms of PSNR. ViLBaS gives an overall delay slightly higher to the one obtained when using OLSR with hop-count. However, ViLBaS obtains a much lower standard deviation, denoting a more stable behaviour. In

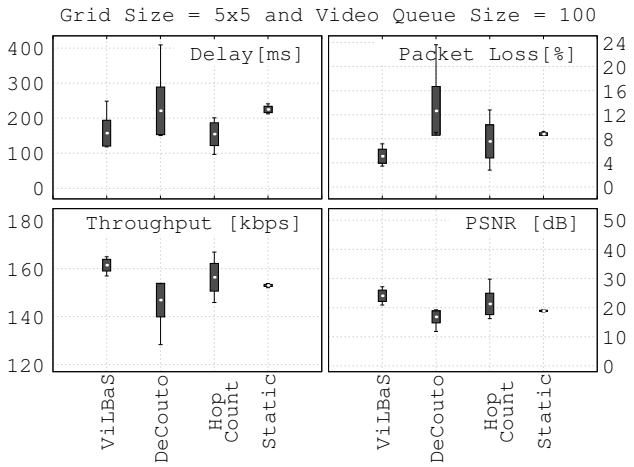

Fig. 12: ViLBaS performance for a queue size of 100 packets in a 5x5 grid topology

TABLE XI: 5x5 Grid topology, Queue size = 100 packets

	Delay [ms]	Packet Loss [%]	Throughput [kbps]	PSNR [dB]	Decrease [%]
ViLBaS	157.40	5.10	161.42	24.07	—
De Couto	221.16	12.63	146.86	16.82	30
Hop Count	154.62	7.55	156.37	21.27	11
Static	225.12	8.78	152.96	18.92	21

TABLE XII: PSNR Values for All Considered Scenarios

	4x4 Grid		5x5 Grid	
	50	100	50	100
ViLBaS	28.77	31.74	24.72	24.07
De Couto	22.56	21.20	17.24	16.82
Hop Count	20.81	22.32	19.90	21.27
Static	22.20	23.12	18.44	18.92

terms of packet loss, ViLBaS keeps this value low at 5.10% losses of all transmitted packets. This value is 32% smaller than the one given by using OLSR with hop-count and is mirrored into the PSNR value. ViLBaS achieves 24.07 dB, which is 11% higher then OLSR with hop-count.

3) Discussion: The overall PSNR values obtained for all scenarios considered are presented in Table XII. Note that for all considered scenarios (i.e. 4x4 and 5x5 grid topologies, and each with 50 and 100 video packets queue lengths, respectively), ViLBaS outperforms the other three solutions considered for comparison: OLSR employing De-Cunto metric, OLSR with hop-count, and the static routing case. This and all previous results show clearly that ViLBaS can provide increased QoS levels for video transmissions to its users over a wireless mesh deployment.

Table XIII presents the average number of packets dropped due to the fact that the wireless interface queue

TABLE XIII: No. of Packets Dropped Due to Full Queue

	4x4 Grid		5x5 Grid	
	50	100	50	100
ViLBaS	37	7	80	72
De Couto	116	136	268	220
Hop Count	145	110	201	156
Static	130	104	180	158

is full, for all considered scenarios. All the values in the table are rounded up. As ViLBaS considers the queue's occupancy to re-route flows, it avoids network congestion and therefore obtains the lowest number of dropped packets. It can be observed that increasing the queue capacity leads to an even smaller amount of lost packets. However, even in this case, ViLBaS detects node congestion and re-routes efficiently video flows, outperforming the other three considered solutions.

4) Overhead Analysis: This section analyses the overhead introduced by ViLBaS in comparison to the other solutions considered. For simplicity, the overhead analysis considers a 16-node grid topology and a 100 second interval.

Hop-count and De Couto metric are both build on top of OLSR, which periodically sends HELLO and TC messages. HELLO messages are sent only to the one-hop neighbours, while TC messages are broadcasted and retransmitted by every node the network. According to RFC 3626 [10], a HELLO message is sent every 2 seconds and a TC message every 5 seconds. Thus, in a 4x4 grid network, 3600 messages are exchanged in 100 seconds.

The *static routing* mechanism incurs no overhead, as the routes discovered initially are kept static. However, this affects the QoS for video deliveries, as the routes do not adapt to the traffic conditions.

ViLBaS, our proposed solution, involves exchanging messages with neighbouring nodes for updating the routes of selected video flows. In our tests, during the simulation period, the largest number of video flows re-routed by ViLBaS was seven, while the smallest number was three. For overhead analysis purposes, we consider the worst case scenario, when ViLBaS re-routes seven flows on the longest path of six nodes. An example of message exchange triggered by the congested node is depicted in Figure 4. The first message is sent by the congested node (i.e. node 6) to the previous node (i.e. node 5). Then, four messages are sent by the previous node (i.e. node 5) to all its one-hop neighbours, followed by four reply messages. Thus, in total nine messages are exchanged at every node, leading to 54 messages for a six-hop path. Considering the worst case scenario of seven video flows being re-routed, this would lead to 378 overhead messages. Comparing with the first two solutions, ViLBaS introduces 90% less overhead.

V. Discussion

ViLBaS Parameters Setting ViLBaS parameter tuning involved multiple tests with two different topologies, two different queue sizes and a wide range of queue occupancy values in order to enable as much as possible solution independence from operational deployment characteristics. When deploying in a completely new topology or with a different buffer size, it is expected that ViLBaS be used with the default parameter settings as identified part of the research reported in this paper. As these parameters values were identified heuristically, ViLBaS does not guarantee an optimum performance, rather than a good one. However, if specific settings of parameters are desired, ViLBaS deployment enables this change to be performed with the corresponding effects in terms of performance. The trade-off between parameter values and performance effects was discussed in section IV.C.

ViLBaS Flexibility and Limitations ViLBaS testing has involved grid topologies for simplicity. However, as the algorithm is generic and in its core does not consider any particular topology-related characteristics, it is expected to perform equally well with any network topology, including with irregular ones. Similarly, testing has involved selection of particular video traffic source-destination pairs. As the algorithm is applied on the bottleneck link, regardless of where the bottleneck gets formed, the selection of the source and destination nodes does not influence algorithm behaviour or its performance. Finally ViLBaS was proposed for scenarios which do not include node mobility, and security aspects were not considered within the scope of this research. Further detailed work is required in order to enable ViLBaS support for both mobility and security.

ViLBaS and Software Defined Networking (SDN) Some of the management operations which make ViLBaS beneficial could also be achievable via a Software Defined Networking (SDN)-based solution. In fact, if SDN is already deployed in a network, it makes sense to also apply the principle behind ViLBaS using an OpenFlow-based approach; however, most networks do not avail from SDN support yet and therefore ViLBaS can be applied immediately, as described in this paper. As a SDN expert and author of a paper which discusses SDN advantages and limitations says, employing SDN is seductive, but challenging [32]. This is as the benefit of employing SDN in terms of operational efficiency is clear; however SDN deployment is slow mostly due to the cost of acquiring both hardware and software SDN relies on. Cisco also highlights the flexibility and agility offered by SDN, but mentions the complexity and cost of deployment, before concluding that we are facing a relative long term transitory process [33]. In this process solutions such as ViLBaS will co-exist with SDN-based approaches in different deployments.

VI. Conclusions and Future Works

This paper addresses the issue of QoS degradation for video deliveries in WMNs caused by overloaded mesh nodes due to unbalanced traffic distribution. The paper presents ViLBaS, a cross-layer load-balancing mechanism, which detects congested mesh nodes and identifies better routes for selected video flows. ViLBaS re-routes these video flows through less congested paths in order to improve resulting service quality levels. The performance evaluation was carried out using the NS-3 simulator and ViLBaS was compared against three representative state of the art solutions for different topologies and video queue sizes. The results obtained indicate that ViLBaS outperforms these other solutions in terms of both QoS and estimated user perceived video quality.

This paper considers video flows only, but future work will extend ViLBaS to take into consideration other classes of traffic, by looking at their specific queues. Future work will also focus on real-life testing replicating the simulation tests and comparison between simulation and real-life testing results.

References

[1] Cisco, "VNI Forecast and Methodology, 2015-2021, White Paper," June 2017. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.pdf

[2] G.-M. Muntean, "Efficient delivery of multimedia streams over broadband networks using qoas," *IEEE Transactions on Broadcasting*, vol. 52, no. 2, pp. 230–235, June 2006.

[3] A. Moldovan, I. Ghergulescu, and C. H. Muntean, "Vqamap: A novel mechanism for mapping objective video quality metrics to subjective mos scale," *IEEE Transactions on Broadcasting*, vol. 62, no. 3, pp. 610–627, Sep. 2016.

[4] ——, "A novel methodology for mapping objective video quality metrics to the subjective mos scale," in *2014 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*, June 2014, pp. 1–7.

[5] N. Murray, B. Lee, Y. Qiao, and G.-M. Muntean, "The impact of scent type on olfaction-enhanced multimedia quality of experience," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 9, pp. 2503–2515, 2017.

[6] S.-K. Ng and W. K. Seah, "Game-theoretic approach for improving cooperation in wireless multihop networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 40, no. 3, pp. 559–574, 2010.

[7] R. Trestian, K. Katrinis, and G. M. Muntean, "Ofload: An openflow-based dynamic load balancing strategy for datacenter networks," *IEEE Transactions on Network and Service Management*, vol. 14, no. 4, pp. 792–803, Dec 2017.

[8] D. S. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.

[9] T. Kim and J. M. Chang, "Qos-aware energy-efficient association and resource scheduling for hetnets," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 1, pp. 650–664, Jan 2018.

[10] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," RFC 3626 (Experimental), Internet Engineering Task Force, Oct. 2003. [Online]. Available: http://www.ietf.org/rfc/rfc3626.txt

[11] L. Tan, Z. Zhu, F. Ge, and N. Xiong, "Utility maximization resource allocation in wireless networks: Methods and algorithms," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 7, pp. 1018–1034, July 2015.

[12] Z. Yuan, G. Ghinea, and G. M. Muntean, "Beyond multimedia adaptation: Quality of experience-aware multi-sensorial media delivery," *IEEE Trans. on Multimedia*, vol. 17, no. 1, pp. 104–117, Jan 2015.

[13] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," in *Proceedings of the 10th annual international conference on Mobile computing and networking*. ACM, 2004, pp. 114–128.

[14] Y. Yang, J. Wang, and R. Kravets, "Designing routing metrics for mesh networks," in *IEEE Workshop on Wireless Mesh Networks (WiMesh)*. Santa Clara, CA, 2005, pp. 1–9.