



Title	A Delay-aware Packet Prioritisation Mechanism for Voice over IP in Wireless Mesh Networks
Authors(s)	Olariu, Cristian, Fitzpatrick, John, Ghamri-Doudane, Yacine, Murphy, Liam, B.E.
Publication date	2016-04-06
Publication information	Olariu, Cristian, John Fitzpatrick, Yacine Ghamri-Doudane, and Liam Murphy B.E. "A Delay-Aware Packet Prioritisation Mechanism for Voice over IP in Wireless Mesh Networks." IEEE, April 6, 2016. https://doi.org/10.1109/WCNC.2016.7564956 .
Conference details	2016 IEEE Wireless Communications and Networking Conference (WCNC), Doha, Qatar 3-6 April 2016
Publisher	IEEE
Item record/more information	http://hdl.handle.net/10197/7656
Publisher's statement	© 2016 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Publisher's version (DOI)	10.1109/WCNC.2016.7564956

Downloaded 2026-05-02 01:14:49

The UCD community has made this article openly available. Please share how this access benefits you. Your story matters! (@ucd_oa)



© Some rights reserved. For more information

A Delay-aware Packet Prioritisation Mechanism for Voice over IP in Wireless Mesh Networks

Cristian Olariu[†], John Fitzpatrick[§], Yacine Ghamri-Doudane^{||} and Liam Murphy[†]

[†]UCD, School of Computer Science, Belfield, Dublin 4, Ireland

[§] Openet, 6 Beckett Way, Park West Business Park, Dublin 12, Dublin, DB, Ireland

^{||} University of La Rochelle, L3i Lab, Av. Michel Crépeau, 17042 La Rochelle CEDEX, France
cristian.olariu@ucd.ie, john.fitzpatrick@openet.com, yacine.ghamri@univ-lr.fr, Liam.Murphy@ucd.ie

Abstract—This work proposes a novel Delay-aware Packet Prioritisation Mechanism (DPPM) to uniformly distribute the Quality of Service (QoS) level across all Voice Over IP (VoIP) calls in a Wireless Mesh Network (WMN). The method prioritises VoIP packets based on the amount of queueing delay that has been accumulated across multiple hops within the WMN. The accumulated queueing delay is piggybacked over every VoIP packet and is used at the enqueueing phase to place more delayed packets towards the head of the queue. This assures higher priority for more delayed VoIP packets over less delayed VoIP packets. The influence of the queueing delay on voice call quality is further reduced by utilising the proposed DPPM in conjunction with WiFi frame aggregation. This conjunction increases the network’s VoIP call capacity, and this is validated through NS-3 simulations.

I. INTRODUCTION

WMNs represent a key wireless access deployment that has been generating a lot of interest over the last number of years. WMNs represent a solution to allow telecoms- or WiFi-operators to provide large areas with wireless data connectivity at relatively low cost. Thanks to minimal cabling required, this type of network is of particular benefit for transient deployments in which the network must be rapidly and cheaply deployed.

WMNs primarily rely on IEEE 802.11 technology, more commonly called WiFi. This choice is motivated by the low cost of hardware and the unlicensed spectrum in which it operates. However, IEEE 802.11 was not initially designed for real-time multimedia applications, such as VoIP and this is a problem that is further exacerbated in a WMN environment due to the multi-hop nature of such a network type.

A large amount of work in the literature is proposing various solutions to improve the voice call quality and the number of supported VoIP calls over a WMN. The majority of these solutions focus on packet aggregation [1, 2, 3, 4, 5, 6, 7], which works by combining multiple small VoIP packets into a single WiFi frame before transmission, hence lowering the number of frame transmissions required, which further reduces the probability of collision.

VoIP call quality is affected by a number of factors such as the voice codec, absolute end-to-end delay (mouth-to-ear delay), packet delay variation (jitter), and packet loss. The existing packet aggregation mechanisms do not take these call quality metrics into account when performing aggregation but

rather aggregate packets on a first-come, first-served basis [1, 2]. The DPPM mechanism proposed in this paper extends these solutions to essentially provide delay-aware aggregation.

In one of our previous works [8] we have shown how our proposed DPPM can be used in conjunction with a Call Admission Control (CAC) mechanism, and in this paper we provide its algorithm and prove its capability in increasing a WMN’s VoIP capacity.

The remainder of this paper is structured as follows. In Section II, the related work is discussed. Section III gives details about the investigated architecture and describes our proposed DPPM. Section IV describes our method used to evaluate VoIP systems. The simulation scenario and results are elaborated in Section V, and the paper is concluded in Section VI.

II. RELATED WORK

VoIP packet aggregation can be performed in two ways, the simplest being an end-to-end approach in which aggregation is done for packets that flow between two specific nodes, such as the ingress and egress points of the WMN. A more complex but more resource efficient approach is to perform aggregation at each hop in the network. A combination of the two methods, called the accretion algorithm, was proposed in [1] and [2]. Aggregation decisions are taken differently depending on whether the current node is a forwarding or an ingress node. Specifically, ingress nodes use a *forced aggregation delay* which allows the build-up of packets in the ingress node’s buffer, whereas the intermediate nodes make better use of natural media access queueing delay for aggregation. Although this approach brings some benefits, we do not consider it in this work as it introduces extra delay at less loaded ingress points. In this work, the hop-by-hop aggregation approach is used in order to provide a larger number of aggregation possibilities which results in increased packet sizes, more efficient physical medium utilisation and ultimately better network capacity.

In [7] the authors investigate another method to increase packet aggregation opportunities. Their focus is on multi-path networks where a routing decision is made based on a set of weights that are assigned to each direct neighbour. In order to increase the aggregation possibilities they propose to scale the weights using a metric dependent on the available queue

space of the queue. While this approach works, it represents another complicated mechanism that operates modifications on the MAC layer and also is based on the somehow unusual assumption that a wireless node creates a queue for each direct neighbour.

The works in [4, 5] investigate the trade-off between the aggregated packet size and the mandatory delay introduced to assure that enough packets are accumulated in the queue for an optimized aggregation process. If the number of packets to be aggregated is not sufficient but a few packets are buffered in the queue long enough, then only packets with a queueing delay higher than a certain threshold are aggregated. This approach is different from the one proposed in our work in at least two ways. Firstly, in our work we consider the cumulative queueing delay that occurred in all previous hops rather than limiting the algorithm to act only on information gathered in the current node. Secondly, in this work the cumulative queueing delay is used at the enqueueing rather than at the aggregation phase.

The authors of [6] propose a different approach in triggering the aggregation process. They argue that aggregation should be triggered based upon the occurrence of an idle medium event; at that point any available packets in the queue can be aggregated and transmitted. The approach in [6] has the advantage of being relatively simple to implement and in the same time minimises the additional delay introduced by packet aggregation. For these reasons this aggregation triggering approach is used by our work too.

Piggybacking delay-related information has been proposed in [9], where the ingress timestamp of a packet is inserted in the TOS field of the IP header. Obtaining a packet's delay using this method requires the nodes to be clock-synchronized which is not trivial. In [9] a GPS solution is suggested, however our method provides the same information without the need of clock synchronisation.

III. THE PROPOSED DELAY-AWARE PACKET PRIORITISATION MECHANISM

A. System Model

The reference WMN topology used throughout this paper is a classical WMN grid topology and comprises three types of nodes:

- the *Mobile Nodes (MNs)* such as smart-phones, laptops, etc.; these nodes follow user movement patterns by roaming the area covered by the WMN. In the scenarios considered here, no direct MN-to-MN communication is considered.
- the *Access Points (APs)* are the nodes comprising the WMN infrastructure. Neighbouring APs are interconnected through one or more mesh interfaces used only for backhaul traffic. The MNs connect to the WMN infrastructure through an additional access interface equipped on each AP.
- the *Network Gateway (NG)* is a WMN node which, besides taking the role of an AP, has also the role of

connecting the WMN infrastructure to the Internet or other external network[s].

The focus of this work is not on the MN-to-AP link but rather on the AP-to-AP links. These links carry the traffic between APs and the network gateway. This is done in a multi-hop fashion among APs, hence a significant delay may build up and packet losses may occur due to bottlenecks and congestion across the WMN.

B. DPPM Algorithm

The network delay of a packet is composed of two parts: a deterministic cumulative delay, typically small and related to packetization and propagation delays, and a stochastic component which is represented by the cumulative queueing delay. The queueing delay is directly related to the queue size, data rate and the Medium Access Control (MAC) mechanism employed; for this reason it can be kept to a minimal value in the case of wired networks, such as ethernet-based networks. However, in the case of 802.11-based wireless networks, the queueing delay is directly influenced by the load and congestion of the wireless medium; the higher the load and congestion level, the higher the queueing delay [3].

In this work, we assume the use of a QoS-enabled WMN in order to maximise voice call quality, i.e. 802.11e Enhanced Distributed Channel Access (EDCA) MAC layer. As such, each mesh node has four different outgoing queues on each of its interfaces. Each of the four queues is assigned to one of the four Access Categories (ACs) defined by the 802.11e protocol [10]. The ACs are: AC_BK for *Background* traffic, AC_BE for *Best Effort* traffic, AC_VI for *Video* traffic, and AC_VO for *Voice* traffic. Prior to accessing the medium, these queues are competing internally using their specific WiFi back-off settings, as such there is no need for an actual scheduler and the VoIP queue has higher chances of winning this competition.

The DPPM estimates the cumulative queueing delay for each VoIP packet which passes through the AC_VO queue of an AP. It stores that value in a packet header, that we call a *delay field*. The value of the *delay field* is initialised to zero at the ingress node, i.e. the first AP node which receives the VoIP packet from a user. Just before the AP's scheduler dequeues a VoIP packet for transmission, the current queueing delay is added to the total delay in the *delay field*. Hence, at each AP the *delay field* contains the total delay experienced by the VoIP packet since entering the WMN. This information is then used at each AP to insert an incoming VoIP packet in such a way, that more delayed packets are placed closer to the head of the queue. In this work, Push-In-First-Out (PIFO) queues are used for the AC_VO queues in order to implement our DPPM.

In a standard First-In-First-Out (FIFO) queue, packets are always enqueued at the last position, if space allows, whereas in a PIFO queue, packets are enqueued at a position determined by a comparison criteria. In our case, the criteria for comparison is the queueing delay value found in the *delay field*. *Algorithm 1* shows how the PIFO queue works in the DPPM.

Algorithm 1: Inserting a VoIP packet into the PIFO queue

input: cp (current VoIP packet), Q (destination queue)

```
1 if  $Q$  is empty then
2   enqueue  $cp$ ;
3   RETURN;
4 else
5   for  $p \leftarrow Q.begin()$  to  $Q.end()$  do
6     if  $TS_{WMN}(cp) > TS_{WMN}(p)$  then
7       if  $length(Q) == MAX\_SIZE$  then
8         drop  $Q.end()$ ;
9         insert  $cp$  before this  $p$ ;
10        RETURN;
11      else
12         $p++$ ; // increment queue iterator
13  if  $length(Q) < MAX\_SIZE$  then
14    enqueue  $cp$ , the youngest packet, at  $Q.end()$ ;
15  else
16    drop  $cp$ , the youngest packet;
17  RETURN;
```

The algorithm is executed every time a new packet arrives and receives two parameters as input: the current VoIP packet (cp) that has to be pushed-in, and the VoIP queue (Q) where the cp is placed. Packets already belonging to Q (if any) are denoted with p .

If Q is empty, cp is simply enqueued; else, the insertion position is determined by iterative comparisons between the delay of cp and of p packets via the TS_{WMN} (Time Spent in the Wireless Mesh Network) comparison function. The TS_{WMN} function returns the content of the *delay field* when called for cp . When TS_{WMN} is called for p , it returns the content of the *delay field* to which it adds the queueing delay occurred so far in the current queue.

When cp 's delay is bigger than the delay of a p , then cp is *pushed-in* right before that packet p , but not before Q 's current length is verified; if Q 's current occupancy is equal to its maximum size, then the last packet of Q is dropped. If this *for* loop reaches the end of Q , then, if Q 's current occupancy is less than its maximum allowed size, then cp is added to Q . Otherwise, cp is dropped as Q is full and all its packets are older than cp .

PIFO queues require more computational power during the enqueueing phase than normal FIFO queues. The authors of [11] present a possible implementation of PIFO queues. Their performance evaluation showed that current hardware is able to cope with the increased processing power imposed by PIFO queues. PIFO queues were not attractive in the past as the extra processing power needed would render the equipment financially prohibitive, however this is not the case anymore.

The complexity of our proposed algorithm is $O(N)$, where N is typically 50 [12] for current WiFi drivers implementing the 802.11e scheme. Hence, in the worst case scenario (i.e.

congested network) the algorithm has to make 50 trivial number comparisons. However when the network is not congested, the queue will operate almost empty with sporadic occupancy bursts, hence the algorithm has to make fewer than 50 comparisons most of the time.

There are at least two ways possible for placing the cumulative delay value into a VoIP packet. First option is to modify an existing packet header to accommodate the new *delay field*, or second, creating a new proprietary packet header. The first option is more desirable, and in [9] the last 10 bits of the TOS field of the IP header have been used to accomplish this. This simple solution is also backwards compatible with the legacy standard.

For VoIP, an end-to-end delay of more than 150ms has negative effects on the voice call quality, whereas an end-to-end delay of more than 400ms significantly degrades the conversation interactivity [13]. Considering these values, a *delay field* that can store delay values up to 256ms is well suited. Hence, an eight-bit field is sufficient.

Finally, a note on the complexity of DPPM. Finding the position where a VoIP packet is to be enqueued is an operation of complexity $O(N)$, where N is the total number of packets enqueued. In the worst case scenario (i.e. congested node) the algorithm has to make N trivial number comparisons.

C. Delay-Aware Aggregation

The aggregation mechanism is based on encapsulating multiple packets or frames into a single one. This approach considerably increases the performance and capacity of WMNs by combining frequent transmissions of small-payload packets into less frequent transmissions of a larger payload. Frame aggregation was initially specified by the 802.11e standard [10] and was further improved by the 802.11n standard [14]. An 802.11 MAC frame can carry an Aggregated MAC Service Data Unit (A-MSDU) which encapsulates MAC Service Data Units (MSDUs), each MSDU having its own sub-frame header. An end-to-end or a hop-by-hop aggregation scheme is usually used for aggregation.

In an end-to-end approach, the aggregating node selects from the outgoing queue only those packets having the same destination. The aggregated packet is not de-aggregated until it reaches the destination. This method gives good results when the network is highly loaded [3].

In the hop-by-hop approach, the aggregating node selects from the queue those packets having the same next-hop. At each hop, the aggregated frame is de-aggregated and the resulting packets are again considered individually. This approach increases the complexity and processing load on the mesh nodes, but yields better results considering all possible network load conditions [6]. As the sub-frame headers in an aggregated frame contain the source and destination address, it means that the 802.11 frame aggregation feature supports the hop-by-hop aggregation scheme.

The aggregation mechanism is triggered by monitoring threshold crossing. One possible threshold is the maximum number of bytes an A-MSDU can have. For example, if the

queue has available for aggregation enough packets to fill an A-MSDU, then the process is triggered and the A-MSDU is made available at the moment when the MAC layer senses the wireless medium as idle. Another possible threshold is the moment a countdown timer expires; a new aggregation process is triggered regardless of whether the queue has packets available for aggregation or not.

In this work we use as aggregation trigger the indication that the MAC layer is allowed to begin a transmission. When the event occurs, the packet scheduler prepares the aggregated packet from the packets available at that moment in the queue. VoIP packets with the same next hop are collected from the PIFO queue and placed into an A-MSDU. Since the PIFO queue is sorted based on the cumulative queueing delay, having the oldest packet as head of the queue, the more delayed packets are always sent first.

IV. VOIP SYSTEMS EVALUATION

A. VoIP Call Quality Assessment

The most popular model to estimate speech assessment is the E-Model (ITU-T G.107 [13]) which is based on network transmission parameters. The quality assessment of video streams [15] uses a similar approach.

The E-Model is widely accepted as an accurate tool for transmission network planning, and operates under the assumption that perceived quality impairments are additive. By taking into account the influence of the network impairments and the voice codec choice, the E-Model produces a scalar called the *Transmission Rating Factor (R)*, which according to the work in [16, 17], can be represented by $R = 93.4 - I_d - I_e + A$ for a conversation carried over IP networks. This equation is comprised of the following elements: 93.4 is a constant which represents the signal-to-noise ratio obtained by considering the circuit and room noise; I_d represents the impairments caused by the delay of voice signals such as talker echo, listener echo, and absolute signal delay (mouth-to-ear delay); I_e represents the impairments caused by low bit-rate codecs and packet loss; A is the advantage factor and represents the user's willingness to accept lower call quality in exchange for the advantage of access. In order to obtain the Mean Opinion Score (MOS), a conversion formula is provided in [13] which converts R values to MOS values (Table I).

R-value (lower limit)	MOS (lower limit)	User Satisfaction
90	4.34	Very satisfied
80	4.03	Satisfied
70	3.60	Some users dissatisfied
60	3.10	Many users dissatisfied
50	2.58	Nearly all users dissatisfied

TABLE I: R to MOS correspondence from the E-Model for estimative assessment

B. Speech Model

In a VoIP call, samples from the actual audio signal are processed by the audio codec and placed into the payload

field of IP packets. Some codecs use a mechanism called Voice Activity Detection (VAD) to detect whether the user is speaking or is silent. Usually, during silent periods, a signalling packet is sent to the other party where white noise is played-out until the arrival of packets carrying voice samples.

The ITU-T propose a model able to generate speech events that mimic a real human conversation. The model in ITU-T's P.59 Report [18] is a four state transition model; the states can be: *Single Talk₁* (A-talking, B-silence), *Single Talk₂* (A-silence, B-talking), *Double Talk*, and *Mutual Silence*. The period length of a state and the probabilities determining which state to change to, have been extracted from actual human conversations. This model is implemented and incorporated into NS-3 for this work, in order to generate realistic speech activity for full-duplex calls.

C. VoIP Capacity

VoIP capacity is the maximum number of calls a network can support with guaranteed call quality. The call quality threshold is typically specified as a MOS value. The VoIP capacity analysis requires as input the MOS values reported by all VoIP users' devices. Since two packet streams are required to transport the VoIP packets between the caller and callee, the MOSs and the derived VoIP capacity are estimated on each direction. Considering the WMN user as a reference point, then the VoIP traffic flowing away from the end-user is the uplink traffic and the VoIP traffic flowing towards the WMN end-user is the downlink traffic.

The overall mean quality per call, per call direction, is obtained by computing the mean of the samples ($\mu(MOS)$). The variations of the quality per call, per call direction, are captured by the standard deviation of the samples ($\sigma(MOS)$). The overall mean quality of all calls supported by the WMN is obtained by computing the mean of the means ($\mu(\mu(MOS))$), also known as the grand mean. The variation of the mean quality among all calls is captured by the standard deviation of the means ($\sigma(\mu(MOS))$).

In our opinion, it is important to also consider the mean of the standard deviations ($\mu(\sigma(MOS))$) as this captures the overall variation of all calls' quality. This value is higher than the standard deviation of the means ($\sigma(\mu(MOS))$). This statement is supported by the following arguments: a few users may already experience call quality which falls below the MOS threshold, before the moment when the grand mean of the overall call quality ($\mu(\mu(MOS))$) has reached the threshold; hence we consider that the minimum between the standard deviation of the means ($\sigma(\mu)$) and the mean of the standard deviations ($\mu(\sigma)$) of the call quality should be considered as the MOS threshold of the VoIP capacity. It is worth noting that other works do not consider these metrics and primarily focus only on the mean call quality values ($\mu(\mu(MOS))$).

Furthermore, the VoIP capacity is obtained by monitoring the overall mean quality of all ongoing calls, while the number of calls is linearly increased. The network's VoIP capacity is reached when the estimated call quality falls under

the specified MOS threshold. A common procedure in the literature [3, 1, 6] is to consider the threshold around 3.6 on the MOS scale, or the equivalent of 70 on the R scale.

D. Proposed Evaluation Methodology

There are four possible mechanism combinations which are used for evaluating the effects of our DPPM:

- *A: DPPM Disabled & Frame Aggregation Disabled;*
- *B: DPPM Enabled & Frame Aggregation Disabled;*
- *C: DPPM Disabled & Frame Aggregation Enabled;*
- *D: DPPM Enabled & Frame Aggregation Enabled;*

From these four cases, *A* is the case where neither of the two mechanisms is enabled. This reference case, which shows the influence of a plain 802.11e-based WMN on VoIP calls, is used as the basis for comparison for the other three cases. The comparison is done for the grand mean of the call quality ($\mu(\mu(MOS))$), VoIP capacity, and for the variation of the call quality represented with grey areas around the mean. Specifically, the variations are: $\mu(\mu(MOS)) \pm \mu(\sigma(MOS))$ for the mean of the standard deviations (light grey), and $\mu(\mu(MOS)) \pm \sigma(\mu(MOS))$ for the standard deviation of the means (dark grey).

We will take Figure 1 as reference plot for the following explanations. To determine which grand mean call quality (depicted with m in the plot) is higher, we compared the size of the area between m and the x axis, for cases *B*, *C*, and *D*. The results of the comparisons are depicted beside the label m as percentage of improvement against case *A* which is the base case.

The size of the areas around m are used to determine which variation is smaller. The results of the variation comparisons are depicted beside label $v1$ which represents the $\mu(\mu(MOS)) \pm \mu(\sigma(MOS))$ and beside label $v2$ which represents the $\mu(\mu(MOS)) \pm \sigma(\mu(MOS))$ as percentage of improvement compared to case *A*.

V. SIMULATION

A. Simulation Settings

In Table II, the network topology is a 4 by 4 grid with the Network Gateway (NG) located at one corner of the WMN. The inter-node distance is 125 meters and the physical link data rate is set to 6Mbps between APs. The relatively large inter-node distance ensures that APs can not communicate with diagonally located neighbours. Each AP is equipped with three interfaces: two for AP-to-AP communications and one interface for AP-to-MN communications [20]. The AP-to-AP interfaces are configured such that one interface supports the uplink traffic while the other the downlink traffic.

The default queue size in NS3 is 400 packets per Access Category (AC), however this is not a realistic queue size. In this work a queue size of 50 packets per AC is used. This represents the current default queue size used by the most widespread wireless drivers, *MadWiFi* [21] and *ath5k* [12].

Using any WMN routing protocol would introduce statistical artefacts, hence hindering the drawing of statistically significant conclusions from the results. To address this issue,

Parameter	Value
Simulator	NS-3.10 [19]
Topology	4x4 grid with 125m step, no mobility
WiFi interfaces	2 x 802.11a for AP-to-AP 1 x 802.11g for AP-to-MN
WiFi Data Rate	6 Mbps
MAC layer	CSMA-CA
Propagation Model	LogDistancePropagationLossModel Path Loss Exponent: 3 Reference Distance: 1 (m) Reference Loss: 46.667 (dB)
Error Rate Model	YansErrorRateModel
Remote Station Manager	ConstantRateWifiManager
WiFi interfaces queue size	50 packets
Routing Algorithm	Fixed routes, pre-discovered by OLSR
Aggregation Type	Hop-by-hop
VoIP call	Full-duplex, 120 sec. duration, G.729a+VAD @ 20 millisecond.
Speech Model	ITU-T/P.59 [18]
Call Quality Assessment	E-Model[13], $I_e=11$, $B_{pl}=19$, $A=0$
VoIP capacity threshold	$R=70$ (MOS=3.6)
Simulation seeds	5

TABLE II: Simulation Setup

we kept the routing table entries static after the Optimized Link State Routing Protocol (OLSR) was used for initial route discovery.

One full-duplex VoIP call is established by each MN user. Each call is established between a WMN user and a user located outside of the WMN. No VoIP calls are placed on the NG as these VoIP packets would be directly forwarded to the wired interface's packet queue and would not be transported over the WMN. During each simulation run, there is a ramp up period during which calls are randomly initiated with different starting times to avoid traffic source synchronisation issues, and each call is active for 120 seconds [22].

The simulations are carried out in batches, one simulation batch per number of injected calls. The number of injected calls is incremented from 1 to 100 in steps of one, resulting in a total of 100 simulation batches. In each batch, each of the four mechanism combinations mentioned above is simulated five times, each time with a different simulation seed. Thus the total number of simulation performed in this work is 2000.

There are around 312×10^{15} possible combinations of distributing 100 calls over 15 nodes, and testing all possibilities is not feasible, given the very large amount of time and processing resources necessary. Hence, we chose to use a uniform call distribution, which is realistic and would not bias the results towards the best or the worst case scenario.

B. VoIP Call Quality and VoIP Capacity Results

We present the call quality analysis for the uplink (Figure 1) and downlink (Figure 2) in separate plots, as there is an obvious difference between the two cases, as described below.

Figure 1 depicts the results obtained for the MOS and VoIP capacity on the uplink direction. This shows that the call quality, variations of call quality, and VoIP capacity increased for cases *C* and *D* with respect to case *A*, however there was no significant difference between themselves. This means

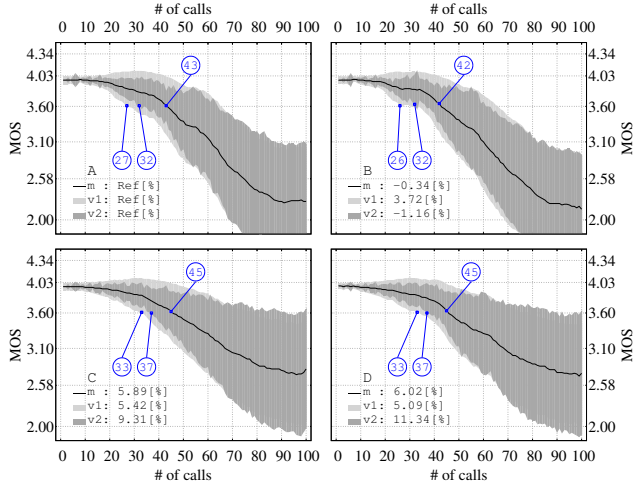


Figure 1: Uplink MOS versus number of injected calls. MOS=3.6 is used as threshold for VoIP capacity estimations. Legend: $m = \mu(\mu(MOS))$, $v1 = m \pm \mu(\sigma(MOS))$ and $v2 = m \pm \sigma(\mu(MOS))$

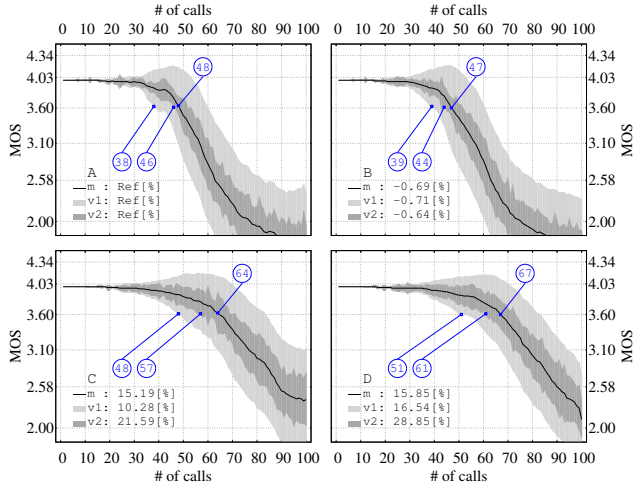


Figure 2: Downlink MOS versus number of injected calls. MOS=3.6 is used as threshold for VoIP capacity estimations. Legend: $m = \mu(\mu(MOS))$, $v1 = m \pm \mu(\sigma(MOS))$ and $v2 = m \pm \sigma(\mu(MOS))$

that, in the uplink direction, the DPPM does not present clear improvement with respect to case C. This is due to the traffic pattern on the uplink, where all VoIP packets are forwarded towards a single node—the network gateway. There are few packets available for aggregation on the APs further away from the gateway, hence only a small improvement can be observed.

However, in the downlink direction as shown in Figure 2, the situation changes. The aggregation mechanism takes advantage of the fact that VoIP packets are always available for aggregation at the gateway. It can be seen that, between cases C and D, the mean call quality remains relatively similar,

however the variations around the mean are smaller when the DPPM is used to enhance the effects of the aggregation mechanism, by about 7% for both $v1$, which represents the $\mu(\mu(MOS)) \pm \mu(\sigma(MOS))$, and $v2$, which represents the $\mu(\mu(MOS)) \pm \sigma(\mu(MOS))$. A smaller variation in case D indicates that the DPPM achieves a more fair distribution of call quality perceived by the users across the network. The other important effect of our proposed DPPM is the increase in VoIP capacity. The VoIP capacity is determined at the intersection of the VoIP capacity threshold, which is set to MOS=3.6, with each of the following: i) the m line, ii) the bottom part of the area representing $v1$, iii) the bottom part of the area representing $v2$. It can be seen in Figure 2 that case D shows an improvement of 3 extra calls, or 6% capacity improvement, over case C.

Both Figures 1 and 2 show that measuring a network’s VoIP capacity can lead to different results, and we consider that the lowest value is the more accurate one. The lowest is the $\mu(\mu(MOS)) - \mu(\sigma(MOS))$ curve which captures the effects of the quality variations across each individual VoIP call, and we propose using it when performing VoIP capacity analysis.

C. Delay, Packet Loss and Jitter Results

Apart from the MOS, other performance metrics were also considered; these were the delay, packet loss and jitter. The means, standard deviation of the means, and mean of the standard deviations for these metrics are shown in Table III for both uplink and downlink.

It can be seen that utilising DPPM in conjunction with WiFi frame aggregation, as shown in case D, results in better overall system performance in terms of variation and overall capacity when compared to the other cases. In general, the variation has improved on the downlink by up to 12% with an average improvement of 8%.

VI. CONCLUSION

The major contribution of this paper is the proposal of a novel packet prioritisation mechanism for improving the quality and capacity of VoIP in WMNs. The absolute network delay of packets traversing the WMN is accurately estimated using DPPM. The delay thus obtained is used to prioritise more delayed over less delayed VoIP packets. The effects of DPPM is improving the system’s capacity and fairness, by providing a more even distribution of packet delay across the WMN.

DPPM performs better on the downlink path of the VoIP packets, as the aggregation mechanism uses larger frames which carry the more delayed VoIP packets in conjunction with being prioritised by DPPM. The variations of the call quality, delay, packet loss, and jitter, decreased in average by about 8% in the case where DPPM is compared to the effects of WiFi frame aggregation, which leads to a fairer distribution of call quality. In addition to the reduction of variation, the achieved VoIP capacity has increased by about 3 calls, which represents an improvement of about 6% in the number of supported VoIP calls when DPPM is compared to WiFi frame

Case	Delay						Packet Loss						Jitter					
	UL			DL			UL			DL			UL			DL		
	<i>m</i>	<i>v</i> ₁	<i>v</i> ₂	<i>m</i>	<i>v</i> ₁	<i>v</i> ₂	<i>m</i>	<i>v</i> ₁	<i>v</i> ₂	<i>m</i>	<i>v</i> ₁	<i>v</i> ₂	<i>m</i>	<i>v</i> ₁	<i>v</i> ₂	<i>m</i>	<i>v</i> ₁	<i>v</i> ₂
B (%)	-3	2	-3	-5	-19	-16	3	13	0	-3	-3	-2	-3	1	-3	-6	-20	-19
C (%)	76	74	71	55	27	40	25	11	11	61	27	38	85	78	74	64	35	40
D (%)	77	75	73	56	33	48	27	11	15	65	35	46	83	78	75	66	41	52

TABLE III: Percentage of improvements for Delay, Packet Loss and Jitter for case B,C, and D, when A is used as reference. Legend: *m* represents the $\mu(\mu(\text{metric}))$, *v1* represents the $\mu(\mu(\text{metric})) \pm \mu(\sigma(\text{metric}))$ and *v2* represents the $\mu(\mu(\text{metric})) \pm \sigma(\mu(\text{metric}))$.

aggregation alone. Compared to the baseline case, which is the scenario where WiFi nodes are neither delay-aware neither are using WiFi aggregation, DPPM increased the overall network's capacity by 13 calls in the worst case, which represents around 34% improvement.

Whilst DPPM improves the delay performance of the competing VoIP traffic, admission control mechanisms are essential to curtail over-subscription of voice traffic in the network as described in [8] and [23].

ACKNOWLEDGEMENTS

This research was partially funded by the European Union through the European Regional Development Fund (ERDF) 2014-2020 programme via Enterprise Ireland (EI) grant IP2014 0344, and through the Marie Curie IAPP programme via grant 230684 CarMesh: Ubiquitous Wireless Mesh Networks for Next-Generation Personal Digital Automotive Services, and partially funded by Science Foundation Ireland (SFI) via grant 08/SRC/I1403 FAME SRC.

REFERENCES

- [1] K. Kim, S. Ganguly, R. Izmailov, and S. Hong, "On Packet Aggregation Mechanisms for Improving VoIP Quality in Mesh Networks," in *Vehicular Technology Conference, 2006. VTC 2006-Spring. IEEE 63rd*, vol. 2, may 2006, pp. 891–895.
- [2] S. Ganguly, V. Navda, K. Kim, A. Kashyap, D. Niculescu, R. Izmailov, S. Hong, and S. Das, "Performance Optimizations for Deploying VoIP Services In Mesh Networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 11, pp. 2147–2158, Nov. 2006.
- [3] D. Niculescu, S. Ganguly, K. Kim, and R. Izmailov, "Performance of VoIP in a 802.11 Wireless Mesh Network," in *INFOCOM 2006. 25th IEEE International Conference on Computer Communications. Proceedings*, april 2006, pp. 1–11.
- [4] N. Bayer, M. de Castro, P. Dely, A. Kassler, Y. Koucheryavy, P. Mitoraj, and D. Staehle, "VoIP Service Performance Optimization in Pre-IEEE 802.11s Wireless Mesh Networks," in *Circuits and Systems for Communications, 2008. ICCSC 2008. 4th IEEE International Conference on*, may 2008, pp. 75–79.
- [5] M. Castro, P. Dely, J. Karlsson, and A. Kassler, "Capacity Increase for Voice over IP Traffic through Packet Aggregation in Wireless Multihop Mesh Networks," in *Future Generation Communication and Networking (FGCN 2007)*, vol. 2, dec. 2007, pp. 350–355.
- [6] K. Lee, S. Yun, I. Kang, and H. Kim, "Hop-by-Hop Frame Aggregation for VoIP on Multi-Hop Wireless Networks," in *Communications, 2008. ICC '08. IEEE International Conference on*, may 2008, pp. 2454–2459.
- [7] G. Di Stasi, J. Karlsson, S. Avallone, R. Canonico, A. Kassler, and A. Brunstrom, "Combining multi-path forwarding and packet aggregation for improved network performance in wireless mesh networks," *Computer Networks*, vol. 64, pp. 26–37, 2014.
- [8] C. Olariu, J. Fitzpatrick, Y. Ghamri-Doudane, and L. Murphy, "Provisioning call quality and capacity for femtocells over wireless mesh backhaul," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2013 IEEE 24th International Symposium on*. IEEE, 2013, pp. 3071–3076.
- [9] P. Yuan, J. Schlembach, A. Skoe, and E. Knightly, "Design and implementation of scalable edge-based admission control," *Computer Networks*, vol. 37, no. 5, pp. 507–518, 2001.
- [10] "Medium Access Control (MAC) Quality of Service (QoS) Enhancements (Replaced by approved draft 802.11e/D13.0)," *IEEE Std P802.11e/D13.0*, 2005.
- [11] F. Wang and M. Hamdi, "iPIFO: A Network Memory Architecture for QoS Routers," in *High Performance Switching and Routing, 2007. HPSR '07. Workshop on*, 30 2007-june 1 2007, pp. 1–5.
- [12] FOSS Drivers, "ath5k," 2010. [Online]. Available: <http://linuxwireless.org/en/users/Drivers/ath5k>
- [13] ITU-T G.107, "The E-Model, A Computational Model For Use In Transmission Planning," 2005.
- [14] "Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 5: Enhancements for Higher Throughput," *IEEE Std 802.11n-2009 (Amendment to IEEE Std 802.11-2007 as amended by IEEE Std 802.11k-2008, IEEE Std 802.11r-2008, IEEE Std 802.11y-2008, and IEEE Std 802.11w-2009)*, pp. 1–565, 29 2009.
- [15] A. Hava, Y. Ghamri-Doudane, G.-M. Muntean, and J. Murphy, "Increasing User Perceived Quality by Selective Load Balancing of Video Traffic in Wireless Networks," 2015.
- [16] R. G. Cole and J. H. Rosenbluth, "Voice over IP Performance Monitoring," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 2, pp. 9–24, Apr. 2001.
- [17] A. D. Clark, P. D. F. Iee *et al.*, "Modeling the effects of burst packet loss and recency on subjective voicing quality," 2001.
- [18] ITU-T P.59, "Artificial Conversational Speech," 1993.
- [19] NS-3 v.10, 2011. [Online]. Available: www.nsnam.org
- [20] A. Hava, G.-M. Muntean, and J. Murphy, "ABI: A mechanism for increasing video delivery quality in multi-radio Wireless Mesh Networks," in *Broadband Multimedia Systems and Broadcasting (BMSB), 2014 IEEE International Symposium on*, June 2014, pp. 1–6.
- [21] MadWifi, "Multiband Atheros driver for WiFi," 2006. [Online]. Available: <http://www.madwifi-project.org>
- [22] R. Birke, M. Mellia, M. Petraccia, and D. Rossi, "Understanding VoIP from backbone measurements," in *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*. IEEE, 2007, pp. 2027–2035.
- [23] C. Olariu, M. Foghlu, P. Perry, and L. Murphy, "VoIP Quality Monitoring in LTE Femtocells," in *Integrated Network Management (IM), 2011 IFIP/IEEE International Symposium on*. IEEE, 2011, pp. 501–508.